Package: hwep (via r-universe)

October 29, 2026

Title Hardy-Weinberg Equilibrium in Polyploids

Version 2.0.3

Description Inference concerning equilibrium and random mating in autopolyploids. Methods are available to test for equilibrium and random mating at any even ploidy level (>2) in the presence of double reduction at biallelic loci. For autopolyploid populations in equilibrium, methods are available to estimate the degree of double reduction. We also provide functions to calculate genotype frequencies at equilibrium, or after one or several rounds of random mating, given rates of double reduction. The main function is hwefit(). This material is based upon work supported by the National Science Foundation under Grant No. 2132247. The opinions, findings, and conclusions or recommendations expressed are those of the author and do not necessarily reflect the views of the National Science Foundation. For details of these methods, see Gerard (2023a) <doi:10.1111/biom.13722> and Gerard (2023b) <doi:10.1111/1755-0998.13856>.

License GPL (>= 3)

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

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

Encoding UTF-8

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0), covr, knitr, rmarkdown, ggplot2

Config/testthat/edition 3

Imports bridgesampling, doFuture, doRNG, foreach, future, iterators, methods, pracma, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), rstantools (>= 2.2.0), tensr, updog

VignetteBuilder knitr

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

Biarch true

Depends R (>= 3.4.0)

SystemRequirements GNU make

NeedsCompilation yes

Author David Gerard [aut, cre] (ORCID:
<https://orcid.org/0000-0001-9450-5023>), NSF DBI 2132247 [fnd]
(https://www.nsf.gov/awardsearch/showAward?AWD_ID=2132247)

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

Config/pak/sysreqs make

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

Date/Publication 2025-07-23 17:50:02 UTC

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

RemoteRef HEAD

RemoteSha 343f907910ceeec3eba4bfc68b3f088937f9b48b

Contents

all_multinom . . . 3
ddirmult . . . 3
dgamete . . . 4
drbounds . . . 5
f1dr . . . 6
freqnext . . . 7
gibbs_gl . . . 8
gibbs_gl_alt . . . 9
gibbs_known . . . 10
gsegmat . . . 11
gsegmat_symb . . . 12
hweboot . . . 13
hwefit . . . 14
hwefreq . . . 16
hwelike . . . 17
hwenodr . . . 18
hweustat . . . 19
menbayesgl . . . 20
p_from_alpha . . . 22
qqpvalue . . . 23
rmbayes . . . 24
rmbayesgl . . . 25
rmlike . . . 28
simgl . . . 29
ts_bands . . . 30
zsegarray . . . 31
zygdist . . . 32

Index **34**

`all_multinom` *Get every possible non-negative tuple with of a given sum.*

Description

The total number of rows is `choose(n = k + n - 1, k = k - 1)`. This function uses recursion, so is not the most efficient.

Usage

```
all_multinom(n, k)
```

Arguments

`n`	Number of indistinguishable balls.
`k`	Number of distinguishable bins.

Value

A matrix, rows index different possible multinomial counts, the columns index the bins.

Author(s)

David Gerard

Examples

```
n <- 5
k <- 3
all_multinom(n = n, k = k)
choose(n = n + k - 1, k = k - 1)
```

`ddirmult` *PMF of Dirichlet-multinomial distribution*

Description

PMF of Dirichlet-multinomial distribution

Usage

```
ddirmult(x, alpha, lg = FALSE)
```

Arguments

x	The vector of counts.
alpha	The vector of concentration parameters.
lg	A logical. Should we log the density (TRUE) or not (FALSE)?

Author(s)

David Gerard

Examples

```
ddirmult(c(1, 2, 3), c(1, 1, 1))
ddirmult(c(2, 2, 2), c(1, 1, 1))
```

---

dgamete *Gamete dosage probability*

Description

Estimates the probability of a gamete dosage given the parent dosage (G), the parent ploidy (ploidy), and the double reduction parameter (alpha). This is for biallelic loci.

Usage

```
dgamete(x, alpha, G, ploidy, log_p = FALSE)
```

Arguments

x	A vector of numerics in seq(0, ploidy/2). The dosage of the gametes.
alpha	A numeric vector containing the double reduction parameter(s). This should be a vector of length floor(ploidy/4) where alpha[i] is the probability of exactly i pairs of IBDR alleles being in the gamete. Note that sum(alpha) should be less than 1, as 1 - sum(alpha) is the probability of no double reduction.
G	The dosage of the parent. Should be an integer between 0 and ploidy.
ploidy	The ploidy of the species. This should be an even positive integer.
log_p	A logical. Should we return the log-probability (TRUE) or not (FALSE)? Defaults to FALSE.

Value

A vector of length length(x), containing the (log) probabilities of a gamete carrying a dosage of x from a parent of dosage G who has ploidy ploidy and a double reduction rate alpha.

Author(s)

David Gerard

Examples

```
dgamete(x = 0:2, alpha = 0, G = 2, ploidy = 4)
```

`drbounds` *Upper bounds on rates of double reduction*

Description

Calculates the upper bounds of the double reduction parameters according to the complete equation segregation model. See Huang et. al. (2019) for details.

Usage

```
drbounds(ploidy)
```

Arguments

`ploidy`	The ploidy of the species. Should be even and at least 4.

Value

A vector of length `floor(ploidy/4)`. Element `i` is the upper bound on the probability of `i` pairs of identical-by-double-reduction alleles being in an individual.

Author(s)

David Gerard

References

- Huang, K., Wang, T., Dunn, D. W., Zhang, P., Cao, X., Liu, R., & Li, B. (2019). Genotypic frequencies at equilibrium for polysomic inheritance under double-reduction. *G3: Genes, Genomes, Genetics*, 9(5), 1693-1706. doi:10.1534/g3.119.400132

Examples

```
drbounds(4)
drbounds(6)
drbounds(8)
drbounds(10)
drbounds(12)
drbounds(14)
drbounds(16)
```

f1dr *Estimate Double Reduction in F1 Populations*

Description

Estimates double reduction in F1 populations by maximum likelihood.

Usage

```
f1dr(nvec, G1, G2)
```

Arguments

nvec	A vector containing the observed genotype counts, where nvec[[i]] is the number of individuals with genotype i-1. This should be of length ploidy+1.
G1	The dosage of parent 1. Should be an integer between 0 and ploidy.
G2	The dosage of parent 2. Should be an integer between 0 and ploidy.

Value

A list with some or all of the following elements:

alpha A vector of numerics of length floor(ploidy / 4), the estimated double reduction rate.

llike The final log-likelihood.

Author(s)

David Gerard

See Also

zygdist() for calculating the probability of offpring genotypes given parental genotypes and the double reduction rate.

Examples

```
set.seed(1)
size <- 100
qvec <- zygdist(alpha = 0.1, G1 = 2, G2 = 2, ploidy = 4)
nvec <- c(stats::rmultinom(n = 1, size = size, prob = qvec))
f1dr(nvec = nvec, G1 = 2, G2 = 2)
```

`freqnext` *Update genotype frequencies after one generation*

Description

After one generation of random mating, update the genotype frequencies.

Usage

```
freqnext(freq, alpha, segmat = NULL, more = FALSE, check = TRUE)
```

Arguments

`freq`	The current genotype frequencies. This should be a vector of length K+1, where K is the ploidy of the species. `freq[i]` could contain the proportion of individuals that have genotype `i-1`.
`alpha`	A numeric vector containing the double reduction parameter(s). This should be a vector of length `floor(ploidy/4)` where `alpha[i]` is the probability of exactly `i` pairs of IBDR alleles being in the gamete. Note that `sum(alpha)` should be less than 1, as `1 - sum(alpha)` is the probability of no double reduction.
`segmat`	You can provide your own segregation matrix. `segmat[i, j]` is the probability that a parent with dosage `i-1` produces a gamete with dosage `j-1`.
`more`	A logical. Should we return more output (`TRUE`) or less (`FALSE`). See the Value section for details.
`check`	Should we correct for minor numerical issues? Defaults to `TRUE`.

Value

If `more = FALSE`, then returns a vector of length `length(freq)` that contains the updated genotype frequencies after one generation of random mating. If `more = TRUE`, then returns a list with these genotype frequencies (`q`) as well as the parental gamete frequencies (`p`).

Author(s)

David Gerard

Examples

```
freq <- c(0.5, 0, 0, 0, 0.5)
freqnext(freq = freq, alpha = 0)
```

gibbs_gl *Gibbs sampler under random mating using genotype log-likelihoods.*

Description

Gibbs sampler under random mating using genotype log-likelihoods.

Usage

```
gibbs_gl(
  gl,
  alpha,
  B = 10000L,
  T = 1000L,
  more = FALSE,
  lg = FALSE,
  verbose = TRUE
)
```

Arguments

gl	The matrix of genotype log-likelihoods. The columns index the dosages and the rows index the individuals. gl[i,j] is the genotype log-likelihood for individual i at dosage j. It is assumed that natural log is used.
alpha	Vector of hyperparameters for the gamete frequencies. Should be length (x.length() - 1) / 2 + 1.
B	The number of sampling iterations.
T	The number of burn-in iterations.
more	A logical. Should we also return posterior draws (TRUE) or not (FALSE).
lg	Should we return the log marginal likelihood (true) or not (false).
verbose	A logical. Should we print the progress?

Value

A list with some or all of the following elements

- mx: The estimate of the marginal likelihood
- p_tilde: The value of p used to evaluate the posterior density.
- p: The samples of the gamete frequencies
- z: The samples of the individual genotypes
- post: The samples of the full conditionals of p_tilde.

Author(s)

David Gerard

Examples

```
set.seed(1)
ploidy <- 8

## Simulate under the null
p <- stats::runif(ploidy / 2 + 1)
p <- p / sum(p)
q <- stats::convolve(p, rev(p), type = "open")
nvec <- c(stats::rmultinom(n = 1, size = 100, prob = q))
gl <- simgl(nvec)

gibbs_gl(gl = gl, alpha = rep(1, ploidy / 2 + 1), lg = TRUE)
```

---

gibbs_gl_alt *Gibbs sampler under the alternative of non-random mating using genotype log-likelihoods.*

Description

Gibbs sampler under the alternative of non-random mating using genotype log-likelihoods.

Usage

```
gibbs_gl_alt(
  gl,
  beta,
  B = 10000L,
  T = 1000L,
  more = FALSE,
  lg = FALSE,
  verbose = TRUE
)
```

Arguments

gl	The matrix of genotype log-likelihoods. The columns index the dosages and the rows index the individuals. `gl[i,j]` is the genotype log-likelihood for individual i at dosage j. It is assumed that natural log is used.
beta	The concentration hyperparameter for the genotype frequencies.
B	The number of sampling iterations.
T	The number of burn-in iterations.
more	A logical. Should we also return posterior draws (TRUE) or not (FALSE).
lg	Should we return the log marginal likelihood (true) or not (false).
verbose	A logical. Should we print the progress?

Value

A list with some or all of the following elements

- mx: The estimate of the marginal likelihood

Author(s)

David Gerard

Examples

```
set.seed(1)
ploidy <- 8

## Simulate under the alternative
q <- stats::runif(ploidy + 1)
q <- q / sum(q)
nvec <- c(stats::rmultinom(n = 1, size = 100, prob = q))
gl <- simgl(nvec)

gibbs_gl_alt(gl = gl, beta = rep(1, ploidy + 1), lg = TRUE)
```

---

gibbs_known *Gibbs sampler under random mating with known genotypes.*

Description

Gibbs sampler under random mating with known genotypes.

Usage

```
gibbs_known(x, alpha, B = 10000L, T = 1000L, more = FALSE, lg = FALSE)
```

Arguments

x	The vector of genotype counts. x(i) is the number of individuals that have genotype i.
alpha	Vector of hyperparameters for the gamete frequencies. Should be length (x.length() - 1) / 2 + 1.
B	The number of sampling iterations.
T	The number of burn-in iterations.
more	A logical. Should we also return posterior draws (TRUE) or not (FALSE).
lg	Should we return the log marginal likelihood (true) or not (false).

Value

A list with some or all of the following elements

- `mx`: The estimate of the marginal likelihood
- `p_tilde`: The value of p used to evaluate the posterior density.
- `p`: The samples of the gamete frequencies
- `post`: The likelihood times prior evaluated at current samples.
- `ptilde_post`: The samples of the full conditionals of p_tilde.

Author(s)

David Gerard

`gsegmat` *Segregation probabilities of gametes*

Description

Produces the segregation probabilities for gamete dosages given parental dosages and the double reduction rate.

Usage

```
gsegmat(alpha, ploidy)
```

Arguments

`alpha`	A numeric vector containing the double reduction parameter(s). This should be a vector of length `floor(ploidy/4)` where `alpha[i]` is the probability of exactly `i` pairs of IBDR alleles being in the gamete. Note that `sum(alpha)` should be less than 1, as `1 - sum(alpha)` is the probability of no double reduction.
`ploidy`	The ploidy of the species. This should be an even positive integer.

Value

A matrix of dimension `ploidy + 1` by `ploidy / 2 + 1`. Element (i, j) is the probability that a parent carrying dosage j - 1 produces a gamete with dosage i - 1.

Author(s)

David Gerard

Examples

```
gsegmat(alpha = NULL, ploidy = 2)

gsegmat(alpha = 1/6, ploidy = 4)

gsegmat(alpha = 0.3, ploidy = 6)

gsegmat(alpha = c(0.35, 0.02), ploidy = 8)

gsegmat(alpha = c(0.4, 0.05), ploidy = 10)
```

---

gsegmat_symb *Symbolic representation of the segregation probability matrix*

Description

Two alleles are identical-by-double-reduction (IBDR) if they originate from the same (by origin) allele in the parent. We let "a" be the probability of zero IBDR alleles, "b" be the probability of one IBDR pair, "c" be the probability of two IBDR pairs, etc...

Usage

```
gsegmat_symb(ploidy, out = c("str", "exp"))
```

Arguments

ploidy	The ploidy of the species
out	Should we return a character matrix ("str") or an expression matrix ("exp")?

Value

A character or expression matrix containing the mathematical form for the segregation matrix. Element (i, j) is the probability a parent with dosage i-1 produces a gamete with dosage j-1.

Author(s)

David Gerard

See Also

gsegmat() for numerical expressions.

Examples

```
gsegmat_symb(4)
gsegmat_symb(6)
gsegmat_symb(8)
```

hweboot *Bootstrap procedure to test for equilibrium*

Description

Iteratively resample individuals/genotypes, calculating the U-statistic for each resample, and use these resamples to test against the null of no equilibrium.

Usage

```
hweboot(n, nboot = 2000, more = FALSE)
```

Arguments

n	One of two forms **A vector of length ploidy + 1** Element i is the number of individuals with genotype i. **A matrix with nsamp rows and ploidy+1 columns** Element (i, j) is the posterior probability that individual i has ploidy j-1.
nboot	The number of bootstrap samples to run.
more	A logical. Should we return the bootstrap replicates (FALSE) or just the p-value, with 95% confidence interval of the p-value (TRUE).

Value

A list with some or all of the following elements

p_hwe The bootstrap p-value against the null of equilibrium.

p_ci The 95% confidence interval of p_hwe.

alpha_boot The bootstrap samples of the double reduction parameter.

u_boot The bootstrap samples of the U-statistic.

Author(s)

David Gerard

Examples

```
set.seed(1)
ploidy <- 6
size <- 100
r <- 0.5
alpha <- 0.1
qvec <- hwefreq(r = r, alpha = alpha, ploidy = ploidy)
nvec <- c(rmultinom(n = 1, size = size, prob = qvec))
bout <- hweboot(n = nvec, more = TRUE, nboot = 1000)
bout$p_hwe
```

```
bout$p_ci
hist(bout$test_boot)
abline(v = bout$test_stat, lty = 2, col = 2)
```

`hwefit` *Equilibrium and random mating estimation and testing for many loci.*

Description

Estimates and tests for either equilibrium or random mating across many loci using `hwelike()`, `hweustat()`, `rmlike()`, `hwenodr()`, or `hweboot()`.

Usage

```
hwefit(
  nmat,
  type = c("ustat", "mle", "rm", "nodr", "boot"),
  effdf = TRUE,
  thresh = 3,
  nboot = 2000,
  verbose = TRUE
)
```

Arguments

`nmat`	A matrix of counts. The rows index the loci and the columns index the genotypes. So `nmat[i, j]` is the number of individuals that have genotype `j-1` at locus `i`. The ploidy is assumed to be `ncol(nmat)-1`.
`type`	The method to use: `"ustat"` U-statistic approach to test for equilibrium and estimate double reduction rates given equilibrium. The default. See `hweustat()`. `"mle"` Maximum likelihood estimation and testing. Only supported for ploidies less than or equal to 10. See `hwelike()`. `"rm"` Testing random mating, and estimating gamete frequencies given random mating. See `rmlike()`. `"nodr"` Testing equilibrium given no double reduction. See `hwenodr()`. `"boot"` Bootstrap approach to test for equilibrium. See `hweboot()`.
`effdf`	A logical. Should we use the effective degrees of freedom? Only applicable if `type = "mle"` or `type = "ustat"`.
`thresh`	A non-negative numeric. The threshold for aggregating genotypes. Only applicable if `type = "mle"`, `type = "ustat"`, or `type = "rm"`.
`nboot`	The number of bootstrap iterations to use if `type = "boot"`.
`verbose`	Should we print more (`TRUE`) or less (`FALSE`)?

Details

We provide parallelization support through the future package.

Value

A data frame. The columns of which can are described in `hwelike()`, `hweustat()`, `rmlike()`, or `hwenodr()`.

Author(s)

David Gerard

Examples

```
## Generate random data
set.seed(5)
ploidy <- 4
nloc <- 100
size <- 1000
r <- 0.25
alpha <- 1/12
qvec <- hwefreq(r = r, alpha = alpha, ploidy = ploidy)
nmat <- t(rmultinom(n = nloc, size = size, prob = qvec))

## Uncomment to run the analysis in parallel on the local
## computer with two workers
# future::plan(future::multisession, workers = 2)
hout <- hwefit(nmat = nmat, type = "ustat")

## Shut down parallel workers
future::plan(sequential)

## Show that p-values are uniform

## QQ-plot on -log10 scale
qqpvalue(pvals = hout$p_hwe, method = "base")

## Kolmogorov-Smirnov Test
stats::ks.test(hout$p_hwe, "qunif")

## Can control for Type I error
mean(hout$p_hwe < 0.05)

## Consistent estimate for alpha
alpha
mean(hout$alpha1)
```

`hwefreq` *Generate HWE genotype frequencies*

Description

Generate genotype frequencies under Hardy-Weinberg equilibrium given the allele frequency of the reference allele (`r`), the double reduction parameter (`alpha`), and the ploidy of the species (`ploidy`).

Usage

```
hwefreq(
  r,
  alpha,
  ploidy,
  niter = 100,
  tol = sqrt(.Machine$double.eps),
  more = FALSE
)
```

Arguments

`r`	The allele frequency of the reference allele.
`alpha`	A numeric vector containing the double reduction parameter(s). This should be a vector of length `floor(ploidy/4)` where `alpha[i]` is the probability of exactly `i` pairs of IBDR alleles being in the gamete. Note that `sum(alpha)` should be less than 1, as `1 - sum(alpha)` is the probability of no double reduction.
`ploidy`	The ploidy of the species. This should be an even positive integer.
`niter`	The maximum number of iterations to simulate.
`tol`	The stopping criterion on the Chi-square divergence between old and new genotype frequencies.
`more`	A logical. Should we return more output (`TRUE`) or less (`FALSE`). See the Value section for details.

Details

If `alpha` is not all 0, then this function repeatedly applies `freqnext()` to simulate genotype frequencies under HWE. Otherwise, it uses `dbinom()`.

Value

If `more = FALSE`, then returns just the genotype frequencies after `niter` generations of random mating. If `more = TRUE`, then returns a list with these genotype frequencies, as well as the parental gamete frequencies.

Author(s)

David Gerard

Examples

```
freq1 <- hwefreq(r = 0.5, alpha = 0, ploidy = 4)
freq2 <- hwefreq(r = 0.5, alpha = 1/6, ploidy = 4)

plot(x = 0:4,
     y = freq1,
     type = "h",
     ylim = c(0, 0.4),
     xlab = "dosage",
     ylab = "Pr(dosage)")
plot(x = 0:4,
     y = freq2,
     type = "h",
     ylim = c(0, 0.4),
     xlab = "dosage",
     ylab = "Pr(dosage)")
```

---

hwelike *Maximum likelihood approach for equilibrium testing and double reduction estimation.*

Description

Genotype frequencies from Huang et al (2019) are used to implement a likelihood procedure to estimate double reduction rates and to test for equilibrium while accounting for double reduction. This approach is only implemented for ploidies 4, 6, 8, and 10.

Usage

```
hwelike(nvec, thresh = 5, effdf = FALSE)
```

Arguments

nvec	A vector containing the observed genotype counts, where nvec[[i]] is the number of individuals with genotype i-1. This should be of length ploidy+1.
thresh	The threshold for ignoring the genotype. We keep genotypes such that nvec >= thresh. Setting this to 0 uses all genotypes.
effdf	A logical. Should we use the ad-hoc "effective degrees of freedom" (TRUE) or not (FALSE)?

Value

A list with some or all of the following elements:

alpha The estimated double reduction parameter(s). In diploids, this value is NULL.

r The estimated allele frequency.

`chisq_hwe` The chi-square test statistic for testing against the null of equilibrium.

`df_hwe` The degrees of freedom associated with `chisq_hwe`.

`p_hwe` The p-value against the null of equilibrium.

Author(s)

David Gerard

References

- Huang, K., Wang, T., Dunn, D. W., Zhang, P., Cao, X., Liu, R., & Li, B. (2019). Genotypic frequencies at equilibrium for polysomic inheritance under double-reduction. G3: Genes, Genomes, Genetics, 9(5), 1693-1706. doi:10.1534/g3.119.400132

Examples

```
thout <- hwefreq(alpha = 0.1, r = 0.3, ploidy = 6)
nvec <- c(stats::rmultinom(n = 1, size = 100, prob = thout))
hwelike(nvec = nvec)
```

`hwenodr` *Test for HWE in autopolyploids under the assumption of no double reduction*

Description

We run a likelihood ratio test against the null of no HWE, assuming that there is no double reduction.

Usage

```
hwenodr(nvec)
```

Arguments

`nvec`	A vector containing the observed genotype counts, where `nvec[[i]]` is the number of individuals with genotype `i-1`. This should be of length `ploidy+1`.

Value

A list with some or all of the following elements

`r` The estimated allele frequency.

`chisq_hwe` The chi-square statistic against the null of equilibrium given no double reduction.

`df_hwe` The degrees of freedom associated with `chisq_hwe`.

`p_hwe` The p-value against the null of equilibrium given no double reduction.

Author(s)

David Gerard

Examples

```
set.seed(10)
qvec <- c(0.2, 0.3, 0.4, 0.1)
nvec <- c(stats::rmultinom(n = 1, size = 100, prob = qvec))
hwenodr(nvec = nvec)
```

---

hweustat *U-process minimizer approach to equilibrium testing and double reduction estimation*

Description

Estimates double reduction and tests for equilibrium while accounting for double reduction. It does this using an approach called "U-process minimization", where we minimize a function of a U-statistic that should be 0 at equilibrium given the true double reduction rate.

Usage

```
hweustat(nvec, thresh = NULL, effdf = TRUE)
```

Arguments

nvec	A vector containing the observed genotype counts, where nvec[[i]] is the number of individuals with genotype i-1. This should be of length ploidy+1.
thresh	The threshold for ignoring the genotype. We keep genotypes such that nvec >= thresh. Setting this to 0 uses all genotypes. Setting this to NULL uses a heuristic that works well in practice.
effdf	A logical. Should we use the ad-hoc "effective degrees of freedom" (TRUE) or not (FALSE)?

Details

This is a two-step estimator, where we first obtain a consistent estimate of the double reduction parameter, use this to estimate the covariance of estimators, then use this to obtain our final estimate of the double reduction parameter.

Value

A list with some or all of the following elements:

alpha The estimated double reduction parameter(s). In diploids, this value is NULL.

chisq_hwe The chi-square test statistic for testing against the null of equilibrium.

df_hwe The degrees of freedom associated with chisq_hwe.

p_hwe The p-value against the null of equilibrium.

Author(s)

David Gerard

Examples

```
set.seed(1)
ploidy <- 6
size <- 1000
r <- 0.1
alpha <- 0.1
qvec <- hwefreq(r = r, alpha = alpha, ploidy = ploidy)
nvec <- c(rmultinom(n = 1, size = size, prob = qvec))
hweustat(nvec = nvec)
```

`menbayesgl` *Bayes test for F1/S1 genotype frequencies using genotype likelihoods*

Description

Uses `get_q_array()` from the updog R package to calculate segregation probabilities (assuming no double reduction) and tests that offspring genotypes follow this distribution.

Usage

```
menbayesgl(
  gl,
  method = c("f1", "s1"),
  p1gl = NULL,
  p2gl = NULL,
  lg = TRUE,
  beta = NULL,
  chains = 2,
  cores = 1,
  iter = 2000,
  warmup = floor(iter/2),
  ...
)
```

Arguments

`gl`	A matrix of genotype log-likelihoods. The rows index the individuals and the columns index the genotypes. So `gl[i,k]` is the genotype log-likelihood for individual `i` at dosage `k-1`. We assume the *natural* log is used (base e).
`method`	Should we test for F1 proportions (`"f1"`) or S1 proportions (`"s1"`)?
`p1gl`	A vector of genotype log-likelihoods for parent 1. `p1gl[k]` is the log-likelihood of parent 1's data given their genotype is `k`.

p2gl	A vector of genotype log-likelihoods for parent 2. p2gl[k] is the log-likelihood of parent 2's data given their genotype is k.
lg	A logical. Should we return the log Bayes factor (TRUE) or the Bayes factor (FALSE)?
beta	The concentration hyperparameters of the genotype frequencies under the alternative of no random mating. Should be length ploidy + 1.
chains	Number of MCMC chains. Almost always 1 is enough, but we use 2 as a default to be conservative.
cores	Number of cores to use.
iter	Total number of iterations.
warmup	Number of those iterations used in the burnin.
...	Control arguments passed to sampling().

Author(s)

David Gerard

References

- Gerard D (2023). "Bayesian tests for random mating in polyploids." *Molecular Ecology Resources*, *23*(8), 1812–1822. doi:10.1111/17550998.13856.

Examples

```
## Not run:
set.seed(1)
ploidy <- 4

## Simulate under the null ----
q <- updog::get_q_array(ploidy = 4)[3, 3, ]

## See BF increases
nvec <- c(stats::rmultinom(n = 1, size = 10, prob = q))
gl <- simgl(nvec = nvec)
menbayesgl(gl = gl, method = "f1")

nvec <- c(stats::rmultinom(n = 1, size = 100, prob = q))
gl <- simgl(nvec = nvec)
menbayesgl(gl = gl, method = "f1")

nvec <- c(stats::rmultinom(n = 1, size = 1000, prob = q))
gl <- simgl(nvec = nvec)
menbayesgl(gl = gl, method = "f1")

## Simulate under the alternative ----
q <- stats::runif(ploidy + 1)
q <- q / sum(q)

## See BF decreases
```

```
nvec <- c(stats::rmultinom(n = 1, size = 10, prob = q))
gl <- simgl(nvec = nvec)
menbayesgl(gl = gl, method = "f1")

nvec <- c(stats::rmultinom(n = 1, size = 100, prob = q))
gl <- simgl(nvec = nvec)
menbayesgl(gl = gl, method = "f1")

nvec <- c(stats::rmultinom(n = 1, size = 1000, prob = q))
gl <- simgl(nvec = nvec)
menbayesgl(gl = gl, method = "f1")


## End(Not run)
```

`p_from_alpha` *Obtain gamete frequencies at equilibrium given rates of double reduction.*

Description

Given the rate of double reduction and the major allele frequency, this function will calculate the gametic frequencies.

Usage

```
p_from_alpha(alpha, p, ploidy)
```

Arguments

`alpha`	A numeric vector containing the double reduction parameter(s). This should be a vector of length `floor(ploidy/4)` where `alpha[i]` is the probability of exactly `i` pairs of IBDR alleles being in the gamete. Note that `sum(alpha)` should be less than 1, as `1 - sum(alpha)` is the probability of no double reduction.
`p`	The allele frequency of the major allele.
`ploidy`	The ploidy of the species.

Value

A numeric vector of length `ploidy / 2 + 1`, where element `i` is the probability that a gamete carries `i-1` copies of the major allele.

Author(s)

David Gerard

Examples

```
p_from_alpha(0.2, 0.5, 4)
```

`qqpvalue` *QQ-plot for p-values*

Description

This will create a QQ-plot for p-values, comparing them to a uniform distribution. We make our plot on the -log10 scale. We calculate simultaneous confidence bands by the Tail Sensitive approach of Aldor-Noiman et al (2013).

Usage

```
qqpvalue(
  pvals,
  method = c("ggplot2", "base"),
  band_type = c("ts", "pointwise"),
  conf_level = 0.95,
  return_plot = FALSE
)
```

Arguments

`pvals`	A vector of p-values.
`method`	Should we use base plotting or ggplot2 (if installed)?
`band_type`	Should we use the method of Aldor-Noiman et al (2013) or pointwise based on beta? Pointwise is not recommended since there is strong dependence between order statistics, and if one is beyond the pointwise bands, then likely lots are also beyond them.
`conf_level`	Confidence level for the bands.
`return_plot`	Should we return the plot? Only applicable if `method == "ggplot2"`.

Author(s)

David Gerard

References

- Aldor-Noiman, S., Brown, L. D., Buja, A., Rolke, W., & Stine, R. A. (2013). The power to see: A new graphical test of normality. The American Statistician, 67(4), 249-260.

See Also

- The `qqPlot()` function from the car package.

Examples

```
set.seed(1)
pvals <- runif(100)
qqpvalue(pvals, band_type = "ts", method = "base")

## Not run:
qqpvalue(pvals, band_type = "ts", method = "ggplot2")

## End(Not run)
```

`rmbayes` *Bayes test for random mating with known genotypes*

Description

Bayes test for random mating with known genotypes

Usage

```
rmbayes(
  nvec,
  lg = TRUE,
  alpha = NULL,
  beta = NULL,
  nburn = 10000,
  niter = 10000,
  type = c("auto", "allo")
)
```

Arguments

`nvec`	A vector containing the observed genotype counts, where `nvec[[i]]` is the number of individuals with genotype `i-1`. This should be of length `ploidy+1`.
`lg`	A logical. Should we return the log Bayes factor (`TRUE`) or the Bayes factor (`FALSE`)?
`alpha`	The concentration hyperparameters of the gamete frequencies under the null of random mating. Should be length ploidy/2 + 1.
`beta`	The concentration hyperparameters of the genotype frequencies under the alternative of no random mating. Should be length ploidy + 1.
`nburn`	The number of iterations in the Gibbs sampler to burn-in.
`niter`	The number of sampling iterations in the Gibbs sampler.
`type`	If `alpha` is `NULL`, then the default priors depend on if you have autopolyploids (`"auto"`) or allopolyploids (`"allo"`).

Author(s)

David Gerard

References

- Gerard D (2023). "Bayesian tests for random mating in polyploids." *Molecular Ecology Resources*, *23*(8), 1812–1822. doi:10.1111/17550998.13856.

Examples

```
set.seed(1)
ploidy <- 8

## Simulate under the null
p <- stats::runif(ploidy / 2 + 1)
p <- p / sum(p)
q <- stats::convolve(p, rev(p), type = "open")

## See BF increase
nvec <- c(stats::rmultinom(n = 1, size = 100, prob = q))
rmbayes(nvec = nvec)

nvec <- c(stats::rmultinom(n = 1, size = 1000, prob = q))
rmbayes(nvec = nvec)

nvec <- c(stats::rmultinom(n = 1, size = 10000, prob = q))
rmbayes(nvec = nvec)

## Simulate under the alternative
q <- stats::runif(ploidy + 1)
q <- q / sum(q)

## See BF decrease
nvec <- c(stats::rmultinom(n = 1, size = 100, prob = q))
rmbayes(nvec = nvec)

nvec <- c(stats::rmultinom(n = 1, size = 1000, prob = q))
rmbayes(nvec = nvec)

nvec <- c(stats::rmultinom(n = 1, size = 10000, prob = q))
rmbayes(nvec = nvec)
```

---

rmbayesgl *Bayes test for random mating using genotype log-likelihoods*

Description

Bayes test for random mating using genotype log-likelihoods

Usage

```
rmbayesgl(
  gl,
  method = c("stan", "gibbs"),
  lg = TRUE,
  alpha = NULL,
  beta = NULL,
  type = c("auto", "allo"),
  chains = 2,
  cores = 1,
  iter = 2000,
  warmup = floor(iter/2),
  ...
)
```

Arguments

gl	A matrix of genotype log-likelihoods. The rows index the individuals and the columns index the genotypes. So gl[i,k] is the genotype log-likelihood for individual i at dosage k-1. We assume the *natural* log is used (base e).
method	Should we use Stan ("stan") or Gibbs sampling ("gibbs")?
lg	A logical. Should we return the log Bayes factor (TRUE) or the Bayes factor (FALSE)?
alpha	The concentration hyperparameters of the gamete frequencies under the null of random mating. Should be length ploidy/2 + 1.
beta	The concentration hyperparameters of the genotype frequencies under the alternative of no random mating. Should be length ploidy + 1.
type	If alpha is NULL, then the default priors depend on if you have autopolyploids ("auto") or allopolyploids ("allo").
chains	Number of MCMC chains. Almost always 1 is enough, but we use 2 as a default to be conservative.
cores	Number of cores to use.
iter	Total number of iterations.
warmup	Number of those iterations used in the burnin.
...	Control arguments passed to sampling().

Author(s)

David Gerard

References

- Gerard D (2023). "Bayesian tests for random mating in polyploids." *Molecular Ecology Resources*, *23*(8), 1812–1822. doi:10.1111/17550998.13856.

Examples

```
## Not run:
set.seed(1)
ploidy <- 4

## Simulate under the null ----
p <- stats::runif(ploidy / 2 + 1)
p <- p / sum(p)
q <- stats::convolve(p, rev(p), type = "open")

## See BF increases
nvec <- c(stats::rmultinom(n = 1, size = 10, prob = q))
gl <- simgl(nvec = nvec)
rmbayesgl(gl = gl)
rmbayesgl(gl = gl, method = "gibbs")

nvec <- c(stats::rmultinom(n = 1, size = 100, prob = q))
gl <- simgl(nvec = nvec)
rmbayesgl(gl = gl)
rmbayesgl(gl = gl, method = "gibbs")

nvec <- c(stats::rmultinom(n = 1, size = 1000, prob = q))
gl <- simgl(nvec = nvec)
rmbayesgl(gl = gl)
rmbayesgl(gl = gl, method = "gibbs")

## Simulate under the alternative ----
q <- stats::runif(ploidy + 1)
q <- q / sum(q)

## See BF decreases
nvec <- c(stats::rmultinom(n = 1, size = 10, prob = q))
gl <- simgl(nvec = nvec)
rmbayesgl(gl = gl)
rmbayesgl(gl = gl, method = "gibbs")

nvec <- c(stats::rmultinom(n = 1, size = 100, prob = q))
gl <- simgl(nvec = nvec)
rmbayesgl(gl = gl)
rmbayesgl(gl = gl, method = "gibbs")

nvec <- c(stats::rmultinom(n = 1, size = 1000, prob = q))
gl <- simgl(nvec = nvec)
rmbayesgl(gl = gl)
rmbayesgl(gl = gl, method = "gibbs")


## End(Not run)
```

`rmlike` *Likelihood inference for random mating*

Description

Estimates gamete genotype frequencies using a maximum likelihood approach and runs a likelihood ratio test for random mating.

Usage

```
rmlike(nvec, thresh = 1, nstarts = 10)
```

Arguments

`nvec`	A vector containing the observed genotype counts, where `nvec[[i]]` is the number of individuals with genotype `i-1`. This should be of length `ploidy+1`.
`thresh`	All groups with counts less than `nvec` will be aggregated together.
`nstarts`	The number of random restarts to the EM algorithm. Set this to 0 for only one run.

Details

Let `q` be the genotype frequencies. Let `p` be the gamete frequencies. Then random mating occurs if `q == stats::convolve(p, rev(p), type = "open")`. We test for this hypothesis using likelihood inference, while estimating `p`.

Value

A list with the following elements:

`p` The estimated gamete genotype frequencies. `p[[i]]` is the estimated frequency for gamete genotype `i-1`.

`chisq_rm` The likelihood ratio test statistic for testing against the null of random mating.

`df_rm` The degrees of freedom associated with `chisq_rm`.

`p_rm` The p-value against the null of random mating.

Author(s)

David Gerard

Examples

```
## Randomly generate gamete frequencies
set.seed(1)
ploidy <- 10
pvec <- stats::runif(ploidy / 2 + 1)
pvec <- pvec / sum(pvec)

## Genotype frequencies from gamete frequencies under random mating
qvec <- stats::convolve(pvec, rev(pvec), type = "open")

## Generate data
nvec <- c(stats::rmultinom(n = 1, size = 100, prob = qvec))

## Run rmlike()
rmlike(nvec = nvec)
```

---

simgl *Simulator for genotype likelihoods.*

Description

Uses the updog R package for simulating read counts and generating genotype log-likelihoods.

Usage

```
simgl(
  nvec,
  rdepth = 10,
  od = 0.01,
  bias = 1,
  seq = 0.01,
  ret = c("gl", "gp", "all"),
  est = FALSE,
  ...
)
```

Arguments

nvec	The genotype counts. nvec[k] contains the number of individuals with genotype k-1.
rdepth	The read depth. Lower means more uncertain.
od	The overdispersion parameter. Higher means more uncertain.
bias	The allele bias parameter. Further from 1 means more bias. Must greater than 0.
seq	The sequencing error rate. Higher means more uncertain.
ret	The return type. Should we just return the genotype likelihoods ("gl"), just the genotype posteriors ("gp"), or the entire updog output ("all")

est	A logical. Estimate the updog likelihood parameters while genotype (TRUE) or fix them at the true values (FALSE)? More realistic simulations would set this to TRUE, but it makes the method much slower.
...	Additional arguments to pass to flexdog_full().

Value

By default, a matrix. The genotype (natural) log likelihoods. The rows index the individuals and the columns index the dosage. So gl[i,j] is the genotype log-likelihood for individual i at dosage j - 1.

Author(s)

David Gerard

Examples

```
set.seed(1)
simgl(c(1, 2, 1, 0, 0), model = "norm", est = TRUE)
simgl(c(1, 2, 1, 0, 0), model = "norm", est = FALSE)
```

---

ts_bands *Get simultaneous confidence bands for a uniform QQ-plot*

Description

This will provide 100(1-a)% simultaneous confidence bands for a sample of size n. It does this by the "tail-sensitive" approach of Aldor-Noiman et al (2013), which uses simulated uniform vectors. The number of simulations is controlled by nsamp.

Usage

```
ts_bands(n, nsamp = 1000, a = 0.05)
```

Arguments

n	Sample size.
nsamp	Number of simulation repetitions.
a	The significance level.

Details

The procedure used is described in Aldor-Noiman et al (2013). But note that they have a mistake in their paper. Step (e) of their algorithm on page 254 should be the CDF of the Beta distribution, not the quantile function.

Value

A list of length 3. The `$lower` and `$upper` confidence limits at uniform quantiles `$q`.

Author(s)

David Gerard

References

- Aldor-Noiman, S., Brown, L. D., Buja, A., Rolke, W., & Stine, R. A. (2013). The power to see: A new graphical test of normality. The American Statistician, 67(4), 249-260.

Examples

```
ts <- ts_bands(100)

graphics::plot(x = ts$q,
               y = ts$upper,
               type = "l",
               xlim = c(0, 1),
               ylim = c(0, 1),
               xlab = "Theoretical Quantiles",
               ylab = "Empirical Quantiles")
graphics::lines(x = ts$q, y = ts$lower)
graphics::lines(x = ts$q, y = ts$q, lty = 2)
```

---

zsegarray *Zygote segregation distributions.*

Description

Obtains offspring genotype probabilities given parental probabilities, the ploidy of the species, and the overdispersion parameter, for all possible parental genotypes.

Usage

```
zsegarray(alpha, ploidy)
```

Arguments

`alpha`	A numeric vector containing the double reduction parameter(s). This should be a vector of length `floor(ploidy/4)` where `alpha[i]` is the probability of exactly `i` pairs of IBDR alleles being in the gamete. Note that `sum(alpha)` should be less than 1, as `1 - sum(alpha)` is the probability of no double reduction.
`ploidy`	The ploidy of the species. This should be an even positive integer.

Value

An array of probabilities. Element (i, j, k) contains the probability of offspring dosage k-1 given parental dosages i-1 and j-1.

Author(s)

David Gerard

Examples

```
ploidy <- 10
alpha <- c(0.5, 0.1)
p1 <- 4
p2 <- 3
segarray <- zsegarray(alpha = alpha, ploidy = ploidy)
graphics::plot(x = 0:10,
               y = segarray[p1 + 1, p2 + 1, ],
               type = "h",
               ylab = "Pr(dosage)",
               xlab = "dosage")
graphics::mtext(paste0("P1 dosage = ",
                       p1,
                       ", ",
                       "P2 dosage = ",
                       p2))
```

---

zygdist *Zygote dosage probabilities.*

Description

Calculates the distribution of an offspring dosages given parental dosages (`G1` and `G2`), the ploidy of the species (`ploidy`), and the double reduction parameter (`alpha`).

Usage

```
zygdist(alpha, G1, G2, ploidy)
```

Arguments

`alpha`	A numeric vector containing the double reduction parameter(s). This should be a vector of length `floor(ploidy/4)` where `alpha[i]` is the probability of exactly `i` pairs of IBDR alleles being in the gamete. Note that `sum(alpha)` should be less than 1, as `1 - sum(alpha)` is the probability of no double reduction.
`G1`	The dosage of parent 1. Should be an integer between `0` and `ploidy`.
`G2`	The dosage of parent 2. Should be an integer between `0` and `ploidy`.
`ploidy`	The ploidy of the species. This should be an even positive integer.

Value

A vector of probabilities. The `i`th element is the probability that the offspring will have dosage `i-1`.

Author(s)

David Gerard

Examples

```
zygdist(alpha = c(0.5, 0.1), G1 = 4, G2 = 5, ploidy = 8)
```

Index

all_multinom, 3

dbinom, *16*
ddirmult, 3
dgamete, 4
drbounds, 5

f1dr, 6
flexdog_full, *30*
freqnext, 7, *16*
future, *15*

get_q_array, *20*
gibbs_gl, 8
gibbs_gl_alt, 9
gibbs_known, 10
gsegmat, 11, *12*
gsegmat_symb, 12

hweboot, 13, *14*
hwefit, 14
hwefreq, 16
hwelike, *14, 15*, 17
hwenodr, *14, 15*, 18
hweustat, *14, 15*, 19

menbayesgl, 20

p_from_alpha, 22

qqpvalue, 23

rmbayes, 24
rmbayesgl, 25
rmlike, *14, 15*, 28

sampling, *21*, *26*
simgl, 29

ts_bands, 30

zsegarray, 31
zygdist, *6*, 32