Package ‘HardyWeinberg’
April 28, 2021

Type Package
Title Statistical Tests and Graphics for Hardy-Weinberg Equilibrium
Version 1.7.2
Date 2021-04-28
Author Jan Graffelman [aut, cre],
      Christopher Chang [ctb],
      Xavi Puig [ctb],
      Jan Wigginton [ctb],
      Leonardo Ortolena [ctb],
      William R. Engels [ctb]
Maintainer Jan Graffelman <jan.graffelman@upc.edu>
Depends R (>= 1.8.0), mice, Rsolnp
Description Contains tools for exploring Hardy-Weinberg equilibrium (Hardy, 1908; Wein- 
berg, 1908) <doi:10.1126/science.28.706.49> for bi and multi-allelic ge-
netic marker data. All classical tests (chi-square, exact, likelihood-ratio and permuta-
tion tests) with bi-allelic variants are included in the package, as well as func-
tions for power computation and for the simulation of marker data under equilibrium and dise-
quilibrium. Routines for dealing with markers on the X-chromosome are included (Graffel-
man & Weir, 2016) <doi: 10.1038/hdy.2016.20>, including Bayesian procedures. Some ex-
act and permutation procedures also work with multi-allelic variants. Special test proce-
dures that jointly address Hardy-Weinberg equilibrium and equality of allele frequen-
cies in both sexes are supplied, for the bi and multi-allelic case. Functions for testing equilib-
rium in the presence of missing data by using multiple imputation are also provided. Imple-
ments several graphics for exploring the equilibrium status of a large set of bi-allelic mark-
ers: ternary plots with acceptance regions, log-ratio plots and Q-Q plots.
License GPL (>= 2)
LinkingTo Rcpp
Imports Rcpp
NeedsCompilation yes
Repository CRAN
Date/Publication 2021-04-28 15:30:07 UTC
R topics documented:

HardyWeinberg-package ........................................... 3
af ................................................................. 5
AFTtest ............................................................ 5
agcounts ......................................................... 6
alleles ............................................................ 7
AllelesToTriangular ................................................ 8
Alzheimer .......................................................... 9
dgrafelmanweir.bi .................................................. 10
dlevene ............................................................ 11
dlevene.bi ......................................................... 12
EAFExact ........................................................... 13
fisherz .............................................................. 14
fold ................................................................. 15
GenerateSamples ..................................................... 16
genlabels ........................................................... 17
HapMapCHBChr1 ..................................................... 17
HWABO .............................................................. 18
HWAIC ............................................................... 19
HWAlltests .......................................................... 20
HWAlr ............................................................... 21
HWAlrPlot ........................................................... 22
HWChisq ............................................................ 23
HWChisqMat ........................................................ 25
HWChisqStats ........................................................ 26
HWClo ............................................................... 27
HWClr ............................................................... 28
HWClrPlot ........................................................... 29
HWCondProbAB ....................................................... 30
HWD ................................................................. 31
HWData .............................................................. 32
HWExact ............................................................. 33
HWExactMat ........................................................ 35
HWExactPrevious ..................................................... 36
HWExactStats ........................................................ 37
HWf ................................................................. 39
HWGenotypePlot ..................................................... 40
HWIlr ............................................................... 41
HWIlrPlot ........................................................... 42
HWLindley ........................................................... 43
HLRAIITests ........................................................ 44
HLratio .............................................................. 45
HLRtest ............................................................. 46
HWMissing .......................................................... 47
HWNetwork .......................................................... 49
HWPperm ............................................................ 51
HWPperm.mult ....................................................... 52
Description

Contains tools for exploring Hardy-Weinberg equilibrium (Hardy, 1908; Weinberg, 1908) for bi and multi-allelic genetic marker data. All classical tests (chi-square, exact, likelihood-ratio and permutation tests) with bi-allelic variants are included in the package, as well as functions for power computation and for the simulation of marker data under equilibrium and disequilibrium. Routines for dealing with markers on the X-chromosome are included (Graffelman & Weir, 2016) including Bayesian procedures. Some exact and permutation procedures also work with multi-allelic variants. Special test procedures that jointly address Hardy-Weinberg equilibrium and equality of allele frequencies in both sexes are supplied, for the bi and multi-allelic case. Functions for testing equilibrium in the presence of missing data by using multiple imputation are also provided. Implements several graphics for exploring the equilibrium status of a large set of bi-allelic markers: ternary plots with acceptance regions, log-ratio plots and Q-Q plots.

Details
With function `HWTernaryPlot` one can create ternary plots with acceptance regions for HWE. Several routines implement statistical tests for HWE such as `HWChisq`, `HWExact`, `HWLRatio` and `HWPerm`. Bayesian procedures are available using `HWPosterior`. Akaike information criterion for various scenarios can be calculated with `HWAIC`. Power computations are possible with `HWPower`.

Author(s)

Jan Graffelman

Maintainer: Jan Graffelman <jan.graffelman@upc.edu>

References


Examples

```
library(HardyWeinberg)

# draw random SNPs from a population that is in HWE

set.seed(123)

m <- 100  # number of markers
n <- 100  # sample size

X <- HWData(n,m)
out <- HWTernaryPlot(X,100,region=1,vertex.cex=2,signifcolour=TRUE)
```
Function to compute allele frequencies

**af**

**Description**

Function af computes the allele frequencies for a matrix or a vector containing genotypic compositions.

**Usage**

```r
af(x)
```

**Arguments**

- `x`: a vector or matrix with compositions

**Value**

a vector with allele frequencies

**Author(s)**

Jan Graffelman (jan.graffelman@upc.edu)

**See Also**

maf

**Examples**

```r
X <- as.vector(rmultinom(1,100,c(0.5,0.4,0.1)))
X <- X/sum(X)
print(X)
print(af(X))
```

---

**AFtest**

Exact test of equality of allele frequencies for males and females

**Description**

Function AFtest tests equality of allele frequencies for males and females for bi-allelic marker data by means of a Fisher exact test.

**Usage**

```r
AFtest(x, verbose = TRUE, ...)
```
Arguments
x a vector containing the genotypic counts \( c(A,B,AA,AB,BB) \) for a bi-allelic X-chromosomal markers.
verbose verbose = TRUE prints results, verbose = FALSE is silent.
... additional arguments for function fisher.test.

Details
Function \( \text{AFtest} \) constructs the contingency table of sex by allele, and call \( \text{fisher.test} \) to test for equality of allele frequencies. The test assumes Hardy-Weinberg equilibrium.

Value
AC Two-way table of sex by allele
pval p-value of the test

Author(s)
Jan Graffelman <jan.graffelman@upc.edu>

See Also
HWChisq, HWExact

Examples
rs5968922 <- c(A=392, B=212, AA=275, AB=296, BB=80)
AFtest(rs5968922)

---

agcounts Calculate allele and genotype counts for X-chromosomal markers

Description
Function \( \text{agcounts} \) determines sample size, minor are major allele counts, allele counts in females, numbers of males and females and allele frequencies for a vector of genotypes counts of an X-chromosomal markers.

Usage
agcounts(x, verbose = FALSE)

Arguments
x a vector of X-chromosomal genotype counts (A,B,AA,AB,BB)
verbose print the counts if (verbose = TRUE)
`alleles` 7

### Value

- `n` sample size
- `nA` number of A alleles
- `nB` number of B alleles
- `nf` number of females
- `nm` number of males
- `nAf` number of A alleles in females
- `nBf` number of B alleles in females
- `nt` total number of alleles
- `fAA` number of AA females
- `fAB` number of AB females
- `fBB` number of BB females
- `pA` overall A allele frequency
- `pB` overall B allele frequency

### Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

### See Also

`mac`, `link{maf}`

### Examples

```r
rs5968922 <- c(A=392, B=212, AA=275, AB=296, BB=80)
counts <- agcounts(rs5968922)
```

### Description

Function `alleles` extracts the names of the alleles from a named genotype vector.

### Usage

```r
alleles(x, fromlabels = TRUE)
```

### Arguments

- `x` A named or unnamed genotype vector (e.g. `c(AA=10, AB=20, BB=5)`)
- `fromlabels` extract genotypes from the labels of the vector elements, or from the vector elements themselves.
AllelesToTriangular

Value
A character vector with the alleles

Author(s)
Jan Graffelman (jan.graffelman@upc.edu)

See Also
n.alleles

Examples
x <- c(AA=10,AG=10,GG=10,AT=5)
als.x <- alleles(x)

Description
AllelesToTriangular constructs a lower triangular matrix of genotype counts from one or two vectors of alleles. It is particularly useful to create genotype counts for microsatellite data (STRs).

Usage
AllelesToTriangular(A1, A2 = NULL, given=NULL)

Arguments
A1 The first allele of each individual, or a vector with all alleles, two consecutive ones for each individual.
A2 The second allele of each individual (optional).
given A vector of known alleles (optional). This argument can be used to specify alleles that may not exist in the data.

Details
If the data is a single column vector with two successive alleles for each individual, then specify A1 only. If data consists of two columns, each holding one allele of each individual, then specify A1 and A2. Typical STR data that comes in the format of two repeat lengths for a set of individuals can be transformed into a lower triangular matrix with genotype counts. See the examples below.

Value
A lower triangular matrix with genotype counts.
Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

References


See Also
toTriangular

Examples

data(NistSTRs)
A1 <- NistSTRs[,1]
A2 <- NistSTRs[,2]
GM <- AllelesToTriangular(A1,A2)
print(GM)

---

**Alzheimer**

*Genotype frequencies for 70 SNPs related to Alzheimer’s disease*

Description

The dataframe contains the genotype frequencies MM, Mm and mm for the 70 SNPs for both cases and controls. The data are taken from table 7.11 in Laird & Lange.

Usage

data(Alzheimer)

Format

A data frame containing 70 observations.

Source

Laird, N. M. and Lange, C. Table 7.11, p. 124

References

Calculate Graffelman-Weir exact density for bi-allelic X-chromosomal variant

Description

Function `dgraffelmanweir` calculates the probability $P(N_{AB}=n_{ab}$ and $M_{A}=m_{a}$ and $N_{A}=n_{a}$) for a bi-allelic X-chromosomal variant.

Usage

`dgraffelmanweir(x, y)`

Arguments

- `x`: vector with male genotype counts (A,B)
- `y`: vector with female genotype counts (AA,AB,BB)

Value

A single real number

Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

References


See Also

`HWExact`, `HWExactStats`

Examples

```r
males <- c(A=392, B=212)
females <- c(AA=275, AB=296, BB=80)
prob <- dgraffelmanweir.bi(males, females)
print(prob)
```
dlenevne

Calculate Levene’s exact density for k alleles

Description

Function dlenevne calculates Levene’s exact density for a diploid system with k alleles.

Usage

dlenevne(N)

Arguments

N A lower triangular matrix with genotype counts

Details

The supplied matrix of genotype counts should be triangular, with the homozygote counts on the diagonal, and all heterozygote counts below the diagonal.

Value

a single real number

Author(s)

Jan Graffelman (jan.graffelman@upc.edu)

References


See Also

HWExact

Examples

x <- c(AA=12, AB=19, AC=13, BB=7, BC=5, CC=0)
x <- toTriangular(x)
prob <- dlenevne(x)
print(prob)
dlevene.bi  
*Calculate Levene's density for a bi-allelic variant*

**Description**

Program `dlevene.bi` calculates Levene's density (P(AB|A)) for a bi-allelic variant.

**Usage**

`dlevene.bi(x)`

**Arguments**

- `x` a vector of genotype counts (AA, AB, BB)

**Value**

a single real number

**Author(s)**

Jan Graffelman (jan.graffelman@upc.edu)

**References**


**See Also**

`dlevene`, `HWExact`

**Examples**

```r
x <- c(AA=298, AB=489, BB=213)
prob <- dlevene.bi(x)
print(prob)
```
EAFExact

Exact test for equality of allele frequencies in males and females

Description

EAFExact uses a Fisher Exact test to compare allele frequencies in males and females for variants with k alleles (k >= 2).

Usage

EAFExact(m, f, verbose = TRUE, ...)

Arguments

m  
vector or triangular matrix with male genotype counts
f  
vector or triangular matrix with female genotype counts
verbose  
print output (TRUE) or not (FALSE)
...  
additional arguments for fisher.test

Details

For bi-allelic autosomal variants the genotype counts can be supplied as vectors ((AA,AB,BB) for males, and (AA,AB,BB) for females). For X-chromosomal bi-allelic variants the genotype counts can also supplied as vectors ((A,B) for males, and (AA,AB,BB) for females). For multi-allelic autosomal variants male and genotype counts can be supplied as vectors (AA,AB,AC,BB,BC,CC,...) or as a triangular matrix, where matrix rows and columns are labelled with the allele name (A,B,C,...). For multi-allelic X-chromosomal variants, male genotype counts must be supplied as a vector (A,B,C,...) and female genotype counts must be supplied as a triangular matrix. See the examples below.

Value

pval  
p-value
tab  
table with allele counts

Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

See Also

fisher.test
Examples

#
# bi-allelic autosomal
#

m <- c(AA=60, AB=96, BB=44)
f <- c(AA=44, AB=97, BB=59)
EAFtest <- EAFExact(m, f)

#
# bi-allelic X-chromosomal
#

males <- c(A=392, B=212)
females <- c(AA=275, AB=296, BB=80)
EAFtest <- EAFExact(males, females, verbose=TRUE)

#
# tri-allelic autosomal
#

males <- c(AA=20, AB=52, AC=34, BB=17, BC=51, CC=26)
females <- c(AA=28, AB=55, AC=33, BB=18, BC=50, CC=16)
EAFtest <- EAFExact(males, females, verbose=TRUE)

#
# tri-allelic X-chromosomal
#

males <- c(A=15, B=17, C=24)
females <- toTriangular(c(AA=4, AB=2, AC=13, BB=6, BC=19, CC=4))
EAFtest <- EAFExact(males, females, verbose=TRUE)

fisherz

Fisher’s z transformation

Description

Calculates Fisher’s z transformation for a correlation coefficient

Usage

fisherz(r)

Arguments

r a correlation coefficient
fold

Value
a real number

Author(s)
Jan Graffelman (jan.graffelman@upc.edu)

See Also
cor

Examples
```
r <- 0.5
print(fisherz(r))
```

fold

Fold a square matrix

Description
The function `fold` sums corresponding below and above diagonal elements of a square matrix to form a triangular matrix.

Usage
```
fold(X, lower = TRUE)
```

Arguments

- `X` a square matrix
- `lower` logical. If `lower=TRUE` a lower triangular matrix is formed, if not an upper triangular matrix.

Details
Useful for constructing triangular matrices of genotype counts

Value
A matrix

Author(s)
Jan Graffelman <jan.graffelman@upc.edu>

See Also
`lower.tri`, `upper.tri`
Examples

allelenames <- paste("A",11:13,sep="")
allele1 <- factor(c("A11","A11","A12","A12","A13","A12"),levels=allelenames)
allele2 <- factor(c("A11","A12","A12","A13","A13","A11"),levels=allelenames)
GC <- table(allele1,allele2)
GC <- as.matrix(unclass(GC))
GCF <- fold(GC)

GenerateSamples (Generate genotypic compositions)

Description

GenerateSamples generates all possible genotypic compositions (AA,AB,BB) for a given sample size n.

Usage

GenerateSamples(n = 5)

Arguments

n the desired sample size

Value

returns a matrix with in each row a possible genotypic composition for the given sample size.

Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

Examples

GenerateSamples(5)
genlabels

Label genotype counts of a vector or matrix

Description
Function genlabels sets the names of a vector or matrix of genotype counts.

Usage
```
genlabels(X)
```

Arguments
- `X` a 3 (or 5) element vector with genotype counts, a matrix of genotype counts (3 or 5 columns)

Value
A vector or a matrix

Author(s)
Jan Graffelman (jan.graffelman@upc.edu)

See Also
- HWChisq

Examples
```
x <- c(25,50,25)
x <- genlabels(x)
```

HapMapCHBChr1

Genotype frequencies for 225 SNPs on chromosome 1 of the CHB population.

Description
The dataframe contains the genotype frequencies in generic notation, AA, AB and BB the first 225 polymorphic SNPs without missing data on chromosome 1 of the Han Chinese in Beijing. The data are compiled from the HapMap project, phase 3.2, containing genotype information of 84 individuals.

Usage
```
data(HapMapCHBChr1)
```
Format

A matrix containing 225 rows and 3 columns (AA, AB, BB).

Source


References


HWABO

Estimate allele frequencies and test for Hardy-Weinberg equilibrium with a tri-allelic ABO system.

Description

Function HWABO takes four genotype counts ("A","B","AB","OO") and estimates the three allele frequencies using the EM algorithm.

Usage

HWABO(x, p = c(1/3, 1/3, 1/3), maxiter = 50, tol = 1e-10, verbose = TRUE)

Arguments

x a vector with genotype counts ("A","B","AB","OO").

p a vector with initial allele frequencies (by default (1/3,1/3,1/3)).

maxiter maximum number of iterations.

tol tolerance for convergence, 1e-10 by default

verbose print iteration history or not.

Value

pn vector with estimated allele frequencies.

It.hist iteration history with log-likelihood.

expected expected genotype counts under HWE.

Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

See Also

af
**Examples**

```r
x <- c(fA=182,fB=60,nAB=17,nOO=176)
al.fre <- HWABO(x)
#al2 <- HWABO(x,p=c(0.99,0.01,0.01),maxiter=25)
#al3 <- HWABO(x,p=c(0.01,0.99,0.01),maxiter=25)
#al4 <- HWABO(x,p=c(0.01,0.01,0.99),maxiter=25)
```

---

**HWAIC**

*Compute Akaike’s Information Criterion (AIC) for HWP and EAF models*

---

**Description**

Function `HWAIC` calculates Akaike’s Information Criterion for ten different models that describe a bi-allelic genetic variant: M11: Hardy-Weinberg proportions and equality of allele frequencies in the sexes (HWP & EAF); M12: EAF and HWP in males only; M13: EAF and HWP in females only; M14: EAF and equality of inbreeding coefficients in the sexes (EIC); M15: EAF only; M21: HWP in both sexes; M22: HWP for males only; M23: HWP for females only; M24: EIC only; M25: None of the previous.

**Usage**

```r
HWAIC(x, y, tracing = 0, tol = 0.000001)
```

**Arguments**

- **x**: Male genotype counts (AA,AB,BB)
- **y**: Female genotype counts (AA,AB,BB)
- **tracing**: Activate tracing in the maximization of some likelihoods (0=no tracing; 1:tracing)
- **tol**: tolerance for iterative maximization of some likelihoods

**Details**

The log-likelihood for the six models is calculated. For two models (C and E) this is done numerically using package RSolnp.

**Value**

A named vector containing 6 values for AIC

**Author(s)**

Jan Graffelman <jan.graffelman@upc.edu>
References


See Also

HWLRtest

Examples

males <- c(AA=11,AB=32,BB=13)
females <- c(AA=14,AB=23,BB=11)
stats <- HWAIC(males,females)
print(stats)

---

HWAlltests Perform all tests for Hardy-Weinberg equilibrium

Description

HWAlltests performs all classical frequentists tests for Hardy-Weinberg equilibrium and lists their p-values.

Usage

HWAlltests(x, verbose = TRUE, include.permutation.test = FALSE, x.linked = FALSE)

Arguments

x a vector with a set of genotype counts (AA, AB, BB)
verbose print output if set to TRUE
include.permutation.test turns on the permutation test if set to TRUE
x.linked x.linked=FALSE indicates the marker is autosomal (default), and x.linked=TRUE indicates it resides on the X-chromosome.

Details

By default the permutation test is not performed in order to reduce computing time.

Value

A dataframe with test statistics and p-values.

Author(s)

Jan Graffelman <jan.graffelman@upc.edu>
**HWAlr**

**Description**

HWAlr computes the additive log-ratio transformation for genotype counts of bi-allelic genetic markers.

**Usage**

```r
HWAlr(X, zeroadj = 0.5, denominator = 2)
```

**Arguments**

- `X`: A matrix of genotype counts (columns AA, AB and BB)
- `zeroadj`: A zero adjustment parameter (0.5 by default)
- `denominator`: The genotype count put in the denominator of the log-ratio (1=AA, 2=AB, 3=BB)

**Value**

A matrix or vector of log-ratio coordinates

**Author(s)**

Jan Graffelman (jan.graffelman@upc.edu)

**References**


**See Also**

HWClr, HWIlr
HWAlrPlot

Examples

```r
X <- HWData(100,100)
Y <- HWAlr(X)
```

---

**HWAlrPlot**

*Plot genetic markers in additive log-ratio coordinates*

**Description**

`HWAlrPlot` creates a scatter plot of the log-ratio coordinates of bi-allelic genetic markers. Hardy-Weinberg equilibrium is indicated by a straight line in the plot.

**Usage**

```r
HWAlrPlot(X, zeroadj = 0.5)
```

**Arguments**

- `X`: A matrix of genotype counts (columns AA, AB, BB)
- `zeroadj`: Zero-adjustment parameter. Zero counts in the count matrix are substituted by `zeroadj` which is 0.5 by default.

**Value**

NULL

**Author(s)**

Jan Graffelman (jan.graffelman@upc.edu)

**References**


**See Also**

`HWClrPlot`, `HWIlrPlot`

**Examples**

```r
X <- HWCl(HWData(100,100))
HWAlrPlot(X)
```
**Description**

`HWChisq` performs the chi-square test for Hardy-Weinberg equilibrium both for autosomal and X-chromosomal markers.

**Usage**

```r
HWChisq(X, cc = 0.5, verbose = TRUE, x.linked = FALSE, phifixed = NULL)
```

**Arguments**

- `X` a vector containing the genotypic counts (AA, AB, BB for autosomal markers c(A, B, AA, AB, BB) for X-chromosomal markers).
- `cc` the continuity correction parameter (default `cc = 0.5`).
- `verbose` `verbose = TRUE` prints results, `verbose = FALSE` is silent.
- `x.linked` `x.linked = FALSE` indicates the marker is autosomal (default), and `x.linked = TRUE` indicates it resides on the X-chromosome.
- `phifixed` (For X-chromosomal markers only) `phifixed = NULL` indicates that the fraction of males (females) should be estimated from the data (default). If set to any other value (e.g. `phifixed = 0.5`) then the sample is assumed to come from a population with the specified fraction of males.

**Details**

`HWChisq` does a chi-square test for Hardy-Weinberg equilibrium, and by default applies a continuity correction. For extreme allele frequencies, the continuity correction can lead to excessive type 1 error rates, and is better turned off in that case. The continuity correction can be turned off by specifying `cc = 0`.

`HWChisq` can do the chi-square test for both autosomal and X-chromosomal markers. By setting `x.linked = TRUE` the marker will be assumed to be on the X-chromosome, and the count vector supplied should have 5 elements instead of 3 elements for an autosomal marker. For X-chromosomal markers argument `phifixed` is in general best left to its default value (`NULL`). Only in specific situations where the theoretical population sex ratio is known (e.g. in simulation studies where a universe with known gender ratio is sampled) `phifixed` could be set to the theoretical ratio of interest.

When `alternative` is set to `less`, a one-sided test for against a negative inbreeding coefficient (heterozygote excess) is performed. When `alternative` is set to `greater` a one-sided test for against a positive inbreeding coefficient (lack of heterozygotes) is performed.
**Value**

`HWChisq` returns a list with the components:

- `chisq` value of the chi-square statistic. NA is returned if the marker is monomorphic.
- `pval` p-value of the chi-square test for Hardy-Weinberg equilibrium.
- `D` Half the deviation from Hardy-Weinberg equilibrium for the AB genotype.
- `p` the allele frequency of A.
- `f` the inbreeding coefficient.
- `expected` the expected counts under Hardy-Weinberg equilibrium.

**Author(s)**

Jan Graffelman <jan.graffelman@upc.edu>

**References**


For the chi-square test for X-linked markers:


**See Also**

`HWLratio`, `HWChisqStats`

**Examples**

```r
# Test for an autosomal blood group marker
#
x <- c(MM=298, MN=489, NN=213)
HW.test <- HWChisq(x, verbose=TRUE)

# Same test without continuity correction
#
HW.test <- HWChisq(x, cc=0, verbose=TRUE)

# Test for an X-chromosomal SNP.
#
rs5968922 <- c(A=392, B=212, AA=275, AB=296, BB=80)
HW.test <- HWChisq(rs5968922, cc=0, x.linked=TRUE, verbose=TRUE)
```
Description

HWChisqMat executes the Chisquare test for HWE for each row in a matrix.

Usage

HWChisqMat(X, ...)

Arguments

X          A n times 3 matrix of genotypic counts (AA,AB,BB)
...        extra arguments that are passed on to HWChisq

Value

pvalvec    Vector with the p-values of each test
chisqvec   Vector with the chi-square statistics
Dvec       Vector with deviations from independence

Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

See Also

HWChisq

Examples

X <- HWData(100,10)
colnames(X) <- c("MM","MN","NN")
Results <- HWChisqMat(X)
Output <- cbind(X,Results$chisqvec,Results$pvalvec)
print(Output)
**HWChisqStats**

*Fast computation of chi-square statistics for Hardy-Weinberg equilibrium*

**Description**

`HWChisqStats` is a function for the fast computation of chi-square statistics (or the corresponding p-values) for a large set of bi-allelic markers (typically SNPs).

**Usage**

```r
HWChisqStats(X, x.linked = FALSE, pvalues = FALSE)
```

**Arguments**

- **X**: A matrix with genotype counts, one row per marker. `X` should have 5 columns for an X-chromosomal data set and 3 columns for an autosomal data set.
- **x.linked**: Logical indicating whether the markers are autosomal (`x.linked=FALSE`) or X-chromosomal (`x.linked=TRUE`).
- **pvalues**: Logical indicated whether chi-square statistics should be returned (`pvalues=FALSE`) or whether p-values should be returned (`pvalues=TRUE`).

**Details**

Matrix `X` should strictly comply with the following format. For an autosomal dataset is should contain the 3 genotype counts in order (AA,AB,BB). For an X-chromosomal dataset it should contain the 5 genotype counts in order (A,B,AA,AB,BB) where A and B are the male counts and AA, AB and BB the female counts.

This function was written for speed improvement, and should be much faster than looping over the rows of `X` with `HWChisq`. There is no error checking on the supplied data matrix.

**Value**

A vector of chi-square statistics

**Author(s)**

Jan Graffelman <jan.graffelman@upc.edu>

**References**


**See Also**

`HWChisq`
Examples

# # Autosomal example
#
set.seed(123)
X <- HWData(1000,100)
monom <- (X[,2]==0 & X[,1]==0) | (X[,2]==0 & X[,3]==0)
X <- X[!monom,] # exclude monomorphics
Chisq.stats <- HWChisqStats(X,x.linked=FALSE,pvalues=FALSE)
Chisq.pvals <- HWChisqStats(X,x.linked=FALSE,pvalues=TRUE)

# # X-chromosomal example
#
X <- HWData(1000,100,n.males=50,nA=75,x.linked=TRUE)
Chisq.stats <- HWChisqStats(X,x.linked=TRUE,pvalues=FALSE)
Chisq.pvals <- HWChisqStats(X,x.linked=TRUE,pvalues=TRUE)

HWClo

Convert genotype counts to compositions

Description
Function HWClo divides each row of a matrix by its total, and so produces matrix of compositions.

Usage
HWClo(X)

Arguments
X A matrix of (genotype) counts

Value
A matrix

Author(s)
Jan Graffelman <jan.graffelman@upc.edu>

See Also
HWAlr,HWClr,HWIlr

Examples
X <- HWData(2,100)
Y <- HWClo(X)
HWClr computes the centred log-ratio transformation for genotype counts of bi-allelic genetic markers.

Usage

HWClr(X, zeroadj = 0.5)

Arguments

X
A matrix of genotype counts (columns AA, AB and BB)

zeroadj
A zero adjustment parameter (0.5 by default)

Value

A matrix or vector of log-ratio coordinates

Author(s)

Jan Graffelman (jan.graffelman@upc.edu)

References


See Also

HWAlr, HWIlr

Examples

X <- HWData(100,100)
Y <- HWClr(X)
HWClrPlot  Plot genetic markers in centred log-ratio coordinates

Description

HWClrPlot creates a scatter plot of the centred log-ratio coordinates of bi-allelic genetic markers. Hardy-Weinberg equilibrium is indicated by a straight line in the plot.

Usage

HWClrPlot(X, zeroadj = 0.5)

Arguments

X  A matrix of genotype counts (columns AA, AB, BB)
zeroadj  Zero-adjustment parameter. Zero counts in the count matrix are substituted by zeroadj which is 0.5 by default.

Value

NULL

Author(s)

Jan Graffelman (jan.graffelman@upc.edu)

References


See Also

HWAlrPlot, HWIlrPlot

Examples

X <- HWClo(HWData(100,100))
HWClrPlot(X)
HWCondProbAB

Compute probability of a genotypic sample

Description

Computes the probability of a particular genotypic sample given the allele count, sample size and number of heterozygotes.

Usage

HWCondProbAB(n, nA, nAB)

Arguments

n  n is the total sample size (total number of individuals)
nA nA is the number of A alleles in the sample
nAB nAB is the number of heterozygotes in the sample

Value

p  probability of the particular sample

Author(s)

Jan Graffelman (jan.graffelman@upc.edu)

See Also

HWExact

Examples

x <- c(298,489,213)
names(x) <- c("MM","MN","NN")
n <- sum(x)
nMN <- x[2]
p <- HWCondProbAB(n,nM,nMN)
**HWD**

*Compute disequilibrium statistic D*

---

### Description

Function HWD computes Weir’s disequilibrium coefficient D.

### Usage

```r
HWD(X)
```

### Arguments

- **X**
  
a vector of genotype counts (AA, AB, BB)

### Value

Returns the disequilibrium coefficient

### Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

### References


### See Also

- HWF
- HWChisq

### Examples

```r
x <- c(MM=298, MN=489, NN=213)
D <- HWD(x)
cat("Disequilibrium coefficient: ", D, "\n")
```
HWData  
Generate genetic marker data in or out of Hardy-Weinberg Equilibrium

Description

HWData generates samples of genotypic counts under various schemes. It mainly uses sampling from the multinomial distribution given Hardy-Weinberg allele frequencies.

Usage

```
HWData(nm = 100, n = rep(100, nm), f = rep(0, nm), p = NULL, conditional = FALSE, exactequilibrium = FALSE, pdist = "runif", x.linked = FALSE, nA = NULL, n.males=rep(round(0.5*n),nm), ...)```

Arguments

- **nm**: The number of bi-allelic markers.
- **n**: The sample sizes.
- **f**: The inbreeding coefficients (only for autosomal markers)
- **p**: a vector of allele frequencies
- **conditional**: if TRUE Haldane’s distribution is used for sampling, if FALSE a multinomial distribution is used. Replaces parameter `pfixed` from the previous version of the package
- **exaktequilibrium**: generates data in exact HWE if set to TRUE
- **pdist**: Take a random allele frequency from a uniform or beta distribution of `pfixed` = FALSE and `p` is not given.
- **x.linked**: Simulated autosomal markers (`x.linked`=FALSE, the default) or X-chromosomal markers (`x.linked`=TRUE)
- **nA**: A vector of minor allele counts, one for each marker. If not specified, it will be calculated from `p`
- **n.males**: The number of males (only relevant if `x.linked` = TRUE)
- `...`: specific parameters for the uniform or beta

Details

The `exaktequilibrium` option only takes effects for autosomal markers (`x.linked`=FALSE) and multinomial sampling (`conditional`=FALSE).

Option `pfixed` is deprecated and replaced by `conditional`

`HWData` returns a matrix of genotype counts, `nm` by 3 for autosomal markers or `nm` by 5 for X-chromosomal markers. Output is no longer supplied in the compositional form. Function `HWClo` can be used to convert the genotype counts to a composition.

If the inbreeding coefficient is specified (`f`) it will only take effect for autosomal markers (`x.linked`=FALSE) and multinomial sampling (`conditional`=FALSE).
**Value**

\[ X \]

A matrix containing the genotype counts.

**Author(s)**

Jan Graffelman (jan.graffelman@upc.edu)

**See Also**

HWternaryPlot

**Examples**

```r
nm <- 100
n <- 100
out <- HWData(nm,n)
```

---

**HWExact**

*Exact test for Hardy-Weinberg equilibrium*

**Description**

HWExact performs an exact test for Hardy-Weinberg equilibrium.

**Usage**

```r
HWExact(X, alternative = "two.sided", pvaluetype = "selome", eps=1e-10, x.linked = FALSE, verbose = TRUE)
```

**Arguments**

- **X**
  - vector with the genotype counts AA, AB, BB
- **alternative**
  - two.sided (default) will perform a two-sided test where both an excess and a dearth of heterozygotes count as evidence against HWE. less is a one-sided test where only dearth of heterozygotes counts a evidence against HWE, greater is a one-sided test where only excess of heterozygotes counts as evidence against HWE.
- **pvaluetype**
  - if pvaluetype is set to dost then the p-value of a two-sided test is computed as twice the tail area of a one-sided test. When set to selome, the p-value is computed as the sum of the probabilities of all samples less or equally likely as the current sample. When set to midp, the p-value is computed as half the probability of the current sample + the probabilities of all samples that are more extreme.
- **x.linked**
  - x.linked=FALSE indicates the marker is autosomal (default), and x.linked=TRUE indicates it resides on the X-chromosome.
- **eps**
  - a tolerance that can be set for comparing probabilities in order to include tied outcomes
- **verbose**
  - print results or not.
Details

HWExact uses the recursion equations described by Wigginton et. al. For testing large sets of bi-allelic variants, use the faster code in HWExactStats.

For large samples, HWExact may give the error message: "evaluation nested too deeply: infinite recursion". This can usually be resolved by increasing R’s limit on nested expressions with options(expressions=10000) or a higher limit. With higher limits, the error message “protect(): protection stack overflow” can occur. This error can usually be resolved by increasing R’s protection stack with the command line option --max-ppsize 10000 or higher values. However, with such large samples the exact test will give virtually the same result as a chi-square test, and it may be easier to use HWChisq in these circumstances.

Value

pval      p-value of the exact test
prob     probabilities of all possible samples with the same sample size and minor allele count
pofthesample     probability of the observed sample

Author(s)

Jan Graffelman (jan.graffelman@upc.edu)

References


See Also

HWLratio, HWChisq, HWExactStats

Examples

# Example for an autosomal marker using the standard exact p-value
#
x <- c(298,489,213)
names(x) <- c("MM","MN","NN")
HW.test <- HWExact(x,verbose=TRUE)
#
# Example for an autosomal marker using the mid p-value
#
HW.test <- HWExact(x,verbose=TRUE,pvaluetype="midp")
#
# Example x-linked marker
Description

HWExactMat executes a fast Exact test for HWE for each row in a matrix.

Usage

HWExactMat(X, ...)

Arguments

X A n times 3 matrix of genotypic counts (AA,AB,BB)
... extra arguments that are passed on to HWExact

Value

pvalvec Vector with the p-values of each test

Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

See Also

HWExact

Examples

X <- HWData(100,10)
colnames(X) <- c("MM","MN","NN")
Results <- HWExactMat(X)
Output <- cbind(X,Results$pvalvec)
print(Output)
HWExactPrevious performs an exact test for Hardy-Weinberg equilibrium

**Usage**

HWExactPrevious(X, alternative = "two.sided", pvaluetype = "selome", x.linked = FALSE, verbose = FALSE)

**Arguments**

- **X**: vector with the genotype counts AA, AB, BB
- **alternative**: two.sided (default) will perform a two-sided test where both an excess and a dearth of heterozygotes count as evidence against HWE. less is a one-sided test where only dearth of heterozygotes counts as evidence against HWE. greater is a one-sided test where only excess of heterozygotes counts as evidence against HWE.
- **pvaluetype**: if pvaluetype is set to dost then the p-value of a two-sided test is computed as twice the tail area of a one-sided test. When set to selome, the p-value is computed as the sum of the probabilities of all samples less or equally likely as the current sample. When set to midp, the p-value is computed as half the probability of the current sample + the probabilities of all samples that are more extreme.
- **x.linked**: x.linked=FALSE indicates the marker is autosomal (default), and x.linked=TRUE indicates it resides on the X-chromosome.
- **verbose**: print results or not.

**Details**

HWExactPrevious uses the recursion equations described by Wigginton et. al.

For large samples, HWExactPrevious may give the error message: "evaluation nested too deeply: infinite recursion". This can usually be resolved by increasing R's limit on nested expressions with options(expressions=10000) or a higher limit. With higher limits, the error message "protect(): protection stack overflow" can occur. This error can usually be resolved by increasing R's protection stack with the command line option --max-ppsize 100000 or higher values. However, with such large samples the exact test will give virtually the same result as a chi-square test, and it may be easier to use HWChisq in these circumstances.

**Value**

- **pval**: p-value of the exact test
- **prob**: probabilities of all possible samples with the same sample size and minor allele count
- **pofthesample**: probability of the observed sample
HWExactStats

Description

HWExactStats is a function for the computation of Exact p-values for a large set of bi-allelic markers (typically SNPs).

Usage

HWExactStats(X, x.linked = FALSE, plinkcode = TRUE, midp = FALSE,...)

Arguments

X A matrix with genotype counts, one row per marker. X should have 5 columns for an X-chromosomal data set and 3 columns for an autosomal data set.

x.linked Logical indicating whether the markers are autosomal (x.linked=FALSE) or X-chromosomal (x.linked=TRUE).

plinkcode Logical indicating whether to use faster C++ code from the PLINK software.

midp Logical indicating whether to use the mid p-value for the C++ code or not

... Extra arguments that are passed on to HWExact

Examples

# Example autosomal marker
#
x <- c(298, 489, 213)
names(x) <- c("MM", "MN", "NN")
## Not run: HW.test <- HWExactPrevious(x,verbose=TRUE)
#
# Example x-linked marker
#
rs5968922 <- c(A=392, B=212, AA=275, AB=296, BB=80 )
## Not run: HWExactPrevious(rs5968922,x.linked=TRUE,verbose=TRUE)
Details

Matrix `X` should strictly comply with the following format. For an autosomal dataset it should contain the 3 genotype counts in order (AA,AB,BB). For an X-chromosomal dataset it should contain the 5 genotype counts in order (A,B,AA,AB,BB) where `A` and `B` are the male counts and AA, AB and BB the female counts.

Argument `plinkcode=TRUE` (the default) will use C++ code for faster calculation (functions `SNPHWE2` and `SNPHWEX`) with larger datasets. The C++ code was generously shared by Christopher Chang, and the same code is used in the program PLINK (2.0).

Value

A vector of p-values

Author(s)

Jan Graffelman <jan.graffelman@upc.edu> (R code) and Christopher Chang <chrchang523@gmail.com> (C++ code)

References


See Also

`HWExact`

Examples

```r
# Autosomal example
# set.seed(123)
X <- HWData(1000,100)
monom <- (X[,2]==0 & X[,1]==0) | (X[,2]==0 & X[,3]==0)
X <- X[!monom,] # exclude monomorphics
Exact.pvalues <- HWExactStats(X,x.linked=FALSE)
#
# X-chromosomal example
# X <- HWData(1000,100,n.males=50,nA=75,x.linked=TRUE)
Exact.pvalues <- HWExactStats(X,x.linked=TRUE)
```
Description

**HWf** computes the inbreeding coefficient for sample of genotype counts, or a matrix of genotype counts.

Usage

```R
HWf(X)
```

Arguments

- **X**
  - a vector or matrix of genotype counts (AA, AB, BB)

Details

For monomorphic markers a warning is issued, and the estimate for the inbreeding coefficient is NaN.

Value

- Returns a single inbreeding coefficient (intraclass correlation coefficient), if X is a single sample, or a vector of inbreeding coefficients, if X is a matrix with genotype counts.

Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

References


See Also

```R
HWChisq
```

Examples

```R
# A single sample
#
x <- c(MM=298,MN=489,NN=213)

fhat <- HWf(x)

cat("Inbreeding coefficient: ",fhat,"\n")

#
```
# Multiple samples
#
X <- HWData(nm=100,n=1000)
fhat <- HWf(X)

**HWGenotypePlot**  Scatter plot of the genotype frequencies

**Description**

HWGenotypePlot makes a scatterplots of the AB or BB frequency versus the AA frequency and represents a blue curve indicating the Hardy-Weinberg equilibrium condition.

**Usage**

```r
HWGenotypePlot(X, plottype = 1, xlab = expression(f[AA]), ylab =
ifelse(plottype == 1, expression(f[AB]), expression(f[BB])), asp = 1,
pch = 19, xlim = c(0, 1), ylim = c(0, 1), cex = 1, cex.axis = 2, cex.lab = 2, ...)
```

**Arguments**

- **X**  
  A matrix of genotype counts or frequencies with three columns (AA, AB, BB)

- **plottype**  
  plottype=1 produces a plot of AB versus AA, plottype=2 produced a plot of BB versus AA.

- **xlab**  
  A label for the x axis

- **ylab**  
  A label for the y axis

- **asp**  
  Aaspect ratio (1 by default)

- **pch**  
  Plotting character (19 by default)

- **xlim**  
  Limits for the x axis (0-1 by default)

- **ylim**  
  Limits for the y axis (0-1 by default)

- **cex**  
  Character expansion factor (1 by default)

- **cex.axis**  
  Character expansion factor for the axes (2 by default)

- **cex.lab**  
  Character expansion factor for labels of axis (2 by default)

- **...**  
  Additional arguments for the plot function

**Value**

**NULL**

**Author(s)**

Jan Graffelman <jan.graffelman@upc.edu>
HWIlr

See Also

HWternaryPlot

Examples

```r
n <- 100  # sample size
m <- 100  # number of markers
Xc <- HWCol(HWData(n,m))
HWGenotypePlot(Xc,plottype=1,main="Heterozygote-homozygote scatterplot")
```

HWIlr

*Compute isometric log ratio coordinates.*

Description

HWIlr computes isometric log ratio coordinates for genotypic compositions (AA, AB, BB)

Usage

```r
HWIlr(X, zeroadj = 0.5)
```

Arguments

- `X` A matrix of genotype counts, markers in rows, counts for AA, AB and BB in three columns
- `zeroadj` Adjustment for zeros (0.5 by defaults)

Value

A matrix of log ratio coordinates.

Author(s)

Jan Graffelman (jan.graffelman@upc.edu)

References


See Also

HWAlr,HWClr
**Examples**

\[ X \leftarrow \text{HWData}(100,100) \]
\[ Y \leftarrow \text{HWIlr}(X) \]

---

**HWIlrPlot**  
Plot bi-allelic genetic markers in isometric log ratio coordinates

**Description**

HWIlrPlot makes a scatter plot of the isometric log ratio coordinates for bi-allelic markers.

**Usage**

\[ \text{HWIlrPlot}(X, \text{zeroadj} = 0.5, ...) \]

**Arguments**

- **X**: Matrix of genotype counts, one marker per row, AA, AB and BB in three columns
- **zeroadj**: Adjustment for zero values (0.5 by default)
- **...**: Additional arguments for function `plot`

**Value**

A matrix of log ratio coordinates.

**Author(s)**

Jan Graffelman (jan.graffelman@upc.edu)

**References**


**See Also**

HWAlrPlot, HWClrPlot

**Examples**

\[ X \leftarrow \text{HWClo}(\text{HWData}(100,100)) \]
\[ \text{HWIlrPlot}(X) \]
HWLindley

Calculate a posteriori density for Lindley’s alpha

Description

Function HWLindley calculates the posterior density for disequilibrium measure alpha, as defined by Lindley (1988).

Usage

HWLindley(alphaseq = seq(-3, 3, by = 0.01), x)

Arguments

alphaseq a single value or a sequence of values for alpha
x the genotype count vector in format (AA,AB,BB)

Details

Numerical integration is used to compute the density.

Value

a vector with values of the density for each value in alphaseq

Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

References


See Also

HWPosterior

Examples

x <- c(MM=298,MN=489,NN=213)
post.dens <- HWLindley(seq(-2,2,by=0.01),x)
## Not run:
plot(seq(-2,2,by=0.01),post.dens,type="l")
## End(Not run)

## Not run:
plot(seq(-2,2,by=0.01),post.dens,type="l")
## End(Not run)
Function `HWLRAllTests` performs a set of likelihood ratio tests in relation with Hardy-Weinberg proportions (HWP) and equality of allele frequencies (EAF) for autosomal bi-allelic genetic variants.

**Usage**

```r
HWLRAllTests(x, y)
```

**Arguments**

- `x` Male genotype counts (AA, AB, BB)
- `y` Female genotype counts (AA, AB, BB)

**Details**

Function `HWLRAllTests` calls `HWLRtest` and calculates the p-value of six different tests: 1) joint HWP and EAF (A-F); 2) EAF irrespective of HWP (C-F); 3) HWP irrespective of EAF (D-F); 4) HWP versus EIC (given EAF) (A-B); 5) EIC irrespective of EAF (E-F) and 6) HWP versus EIC. Letters refer to scenarios described by Graffelman & Weir (2018).

**Value**

A named vector with six p-values

**Author(s)**

Jan Graffelman <jan.graffelman@upc.edu>

**References**


**See Also**

`HWLRtest`
Examples

```r
males <- c(AA=11, AB=32, BB=13)
females <- c(AA=14, AB=23, BB=11)
pvalues <- HWLRAllTests(males, females)
print(pvalues)
```

**HWLratio**

Likelihood ratio test for Hardy Weinberg equilibrium

**Description**

HWLratio performs the Likelihood ratio test for Hardy Weinberg equilibrium, both for autosomal and X-chromosomal markers.

**Usage**

```r
HWLratio(X, verbose = TRUE, x.linked = FALSE)
```

**Arguments**

- `X` 
  - `X` a vector containing the genotypic counts (AA, AB, BB).
- `verbose` 
  - `verbose = TRUE` prints results, `verbose = FALSE` is silent.
- `x.linked` 
  - `x.linked = FALSE` indicates the marker is autosomal (default), and `x.linked = TRUE` indicates it resides on the X-chromosome.

**Value**

HWLratio returns a list with the components:

- `Lambda` the likelihood ratio
- `G2` 
  - `-2*log(Lambda)`
- `pval` the p-value

**Author(s)**

Jan Graffelman <jan.graffelman@upc.edu>

**References**


**See Also**

HWChisq
Examples

```r
x <- c(298, 489, 213)
names(x) <- c("MM", "MN", "NN")
HW.test <- HWLratio(x, verbose=TRUE)
  # Test for an X-chromosomal SNP.
  #
  rs5968922 <- c(A=392, B=212, AA=275, AB=296, BB=80)
HW.test <- HWLratio(rs5968922, x.linked=TRUE, verbose=TRUE)
```

The function `HWLRtest` performs a likelihood ratio test comparing two nested scenarios for a bi-allelic genetic variant, distinguishing the two sexes.

Description

Program `HWLRtest` performs a likelihood ratio test comparing two scenarios for an autosomal bi-allelic genetic variant. The scenarios concern Hardy-Weinberg proportions (HWP) and equality of allele frequencies (EAF) in both sexes. The different scenarios are described by Graffelman & Weir (2017).

Usage

```r
HWLRtest(x, y, scene.null = "S1", scene.alt = "S6", verbose = TRUE, tracing = 0)
```

Arguments

- `x`: Male genotype counts
- `y`: Female genotype counts
- `scene.null`: Scenario under the null hypothesis (E.g. "S1")
- `scene.alt`: Scenario under the alternative hypothesis (E.g. "S6")
- `verbose`: print output or not
- `tracing`: Show tracing of the numeric likelihood maximization (1) or not (0).

Details

The different scenarios are indicated with S1, S2, S3, S4, S6 and S6. S1 refers to Hardy-Weinberg proportions and equality of allele frequencies. S2 refers to equality of allele frequencies and equality of inbreeding coefficients for the two sexes. S3 refers to equality of allele frequencies irrespective of HWP. S4 refers to HWP irrespective of allele frequencies. S5 refers to equality of inbreeding coefficients irrespective of allele frequencies. S6 is unrestricted.

Value

- `G2`: Likelihood ratio statistic
- `df`: Degrees of freedom of the likelihood ratio statistic
- `pval`: p-value
**HWMissing**

Test a bi-allelic marker for Hardy-Weinberg equilibrium in the presence of missing genotype information.

**Description**

Function `HWMissing` imputes missing genotype data with a multinomial logit model that uses information from allele intensities and/or neighbouring markers. Multiple imputation algorithms implemented in the Mice package are used to obtain imputed data sets. Inference for HWE is carried out by estimating the inbreeding coefficient or exact p-values for each imputed data set, and by combining all estimates using Rubin’s pooling rules.

**Author(s)**

Jan Graffelman <jan.graffelman@upc.edu>

**References**


**See Also**

`HWAIC`

**Examples**

```r
males <- c(AA=11,AB=32,BB=13)
females <- c(AA=14,AB=23,BB=11)

# # test EAF # #
#
# lr1.out <- HWLRtest(males,females,scene.null="S3",scene.alt="S6")
# # test EIC given EAF # #
#
# lr2.out <- HWLRtest(males,females,scene.null="S2",scene.alt="S3")
# # test HWP versus EIC, given EAF. # #
#
# lr3.out <- HWLRtest(males,females,scene.null="S1",scene.alt="S2")
```
HWMissing

**Usage**

```r
HWMissing(X, imputecolumn = 1, m = 50, coding = c(0,1,2), verbose = FALSE, alpha = 0.05, varest = "oneovern", statistic = "chisquare", alternative = "two.sided", ...)
```

**Arguments**

- `X`: An input data frame. By default, the first column should contain the SNP with missing values.
- `imputecolumn`: Indicates which column of the supplied data frame is to be imputed (by default, the first column, `imputecolumn=1`).
- `m`: The number of imputations (50 by default).
- `coding`: Indicates how the genotype data is coded (e.g. 0 for AA, 1 for AB, and 2 for BB).
- `verbose`: `verbose = TRUE` prints results, `verbose = FALSE` is silent.
- `alpha`: Significance level (0.05 by default) used when computing confidence intervals.
- `varest`: Estimator for the variance of the inbreeding coefficient. `varest="oneovern"` is the default and sets the variance under the null (1/n). `varest="bailey"` uses an approximation (see details).
- `statistic`: If `statistic = "chisquare"` then inbreeding coefficients (equivalent to chisquare statistics) will be computed for each imputed data set and then combined. If `statistic = "exact"` then one-sided exact tests will be computed for each imputed data set and the resulting p-values will be combined.
- `alternative`: `two.sided` (default) will perform a two-sided test where both an excess and a dearth of heterozygotes count as evidence against HWE. `less` is a one-sided test where only dearth of heterozygotes counts as evidence against HWE. `greater` is a one-sided test where only excess of heterozygotes counts as evidence against HWE.
- `...`: Additional options for function `mice` of the Mice package.

**Details**

The function `HWMissing` tests one genetic marker (e.g. a SNP) with missings for HWE. By default, this marker is supposed to be the first column of dataframe `X`. The other columns of `X` contain covariates to be used in the imputation model. Covariates will typically be other, correlated markers or allele intensities of the SNP to be imputed. Covariate markers should be coded as factor variables whereas allele intensities should be numerical variables. By default, a polytomous regression model will be used to impute the missings. If the covariates also contain missings, an imputation method for each column of `X` can be specified by using the method of mice (see example below).

If there are no covariates, missings can be imputed under the MCAR assumption. In that case, missings are imputed by taking a random sample from the observed data. This is what `HWMissing` will do if no covariates are supplied, `X` being a single factor variable.

Several estimators for the variance of the inbreeding coefficient have been described in the literature. The asymptotic variance of the inbreeding coefficient under the null hypothesis is 1/n, and is used if `varest = "oneovern"` is used. This is the recommended option. Alternatively, the approximation described in Weir (p. 66) can be used with `varest = "bailey"`. 
Value

<table>
<thead>
<tr>
<th>Res</th>
<th>A vector with the inbreeding coefficient, a confidence interval for the inbreeding coefficient, a p-value for a HWE test and missing data statistics.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xmat</td>
<td>A matrix with the genotypic composition of each of the m imputed data sets.</td>
</tr>
</tbody>
</table>

Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

References


See Also

HWChisq

Examples

data(Markers)
## Not run:
set.seed(123)
Results <- HWNetwork(Markers[,1],m=50,verbose=TRUE)$Res # no covariates, imputation assuming MCAR.
set.seed(123)
Results <- HWNetwork(Markers[,1:3],m=50,verbose=TRUE)$Res # impute with two allele intensities.
set.seed(123)
Results <- HWNetwork(Markers[,c(1,4,5)],m=50,verbose=TRUE)$Res # impute with two covariate SNPs
## End(Not run)

HWNetwork

Autosomal and X-chromosomal exact tests for HWE via a Network algorithm

Description

Program HWNetwork implements a network algorithm for efficient calculation of exact test p-values in HWE tests with multiple alleles.

Usage

HWNetwork(a1, a2, ma = NULL, fe = NULL, gender = NULL, verbose = TRUE)
Arguments

- `a1` the first allele (expressed as a number)
- `a2` the second allele (expressed as a number; NA if the variant is X chromosomal)
- `ma` alternative format: vector of male X chromosomal allele counts.
- `fe` triangular matrix of female genotype counts
- `gender` gender of the individual (1=male; 2=female)
- `verbose` be silent (`verbose=FALSE`) or informative (`verbose=TRUE`)

Details

Function `HWNetwork` accepts data in two formats. Original genotype data (e.g. repeat numbers of microsatellites) can be supplied, or the data can be supplied in summarized form as a male genotype count vector and a female genotype count matrix. If one of the two male alleles is missing (NA) the variant will taken to be X-chromosomal. If all males have two alleles, the variant will taken to be autosomal.

Value

the exact p-value of the test.

Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

References


See Also

`HWExact`, `HWExactStats`

Examples

```r
# From vectors with counts of genotypes
#
data(TSIXTriAllelics)
ma <- as.matrix(TSIXTriAllelics[1,2:4])
names(ma) <- c("A","B","C")

fe <- TSIXTriAllelics[1,5:10]
names(fe) <- c("AA","AB","AC","BB","BC","CC")
```
**HWPerm**

```r
fe <- toTriangular(fe)
HWNetwork(ma=ma, fe=fe)
```

---

### HWPerm

**Permutation test for Hardy-Weinberg equilibrium**

**Description**

Function **HWPerm** does a permutation test for Hardy-Weinberg equilibrium using a user-supplied test statistic.

**Usage**

```r
HWPerm(x, nperm = 17000, verbose = TRUE, x.linked = FALSE,
FUN = ifelse(x.linked, Chisquare.x, Chisquare), eps = 1e-10, ...)
```

**Arguments**

- **x**: A vector of genotype counts (AA, AB, BB)
- **nperm**: The number of permutations
- **verbose**: `verbose = TRUE` will print results, `verbose = FALSE` is silent.
- **x.linked**: `x.linked = FALSE` indicates the marker is autosomal (default), and `x.linked = TRUE` indicates it resides on the X-chromosome.
- **FUN**: An function call for calculating the test statistic for HWE (see examples below)
- **eps**: Tolerance for comparison of floating point numbers (1e-10 by default)
- **...**: Additional parameters for the function call argument `FUN`

**Details**

The set of alleles for the observed sample is permuted. Consequently, the test is conditional on allele frequency.

**Value**

**HWPerm** returns a list with the components:

- **stat**: value of the chosen test statistic for the observed sample.
- **pval**: p-value of the permutation test.

**Author(s)**

Jan Graffelman &lt;jan.graffelman@upc.edu&gt;

**References**

HWPerm.mult

Permutation tests for variants with multiple alleles

Description

Function `HWPerm.mult` implements permutation tests for Hardy-Weinberg equilibrium for autosomal and X-chromosomal variants.

Usage

```r
HWPerm.mult(x, y = NULL, nperm = 17000, eps = 1e-10, verbose = TRUE, ...)```

Arguments

- `x` vector or triangular matrix with male genotype counts
- `y` vector or triangular matrix with female genotype counts
- `nperm` number of permutations (17.000 by default)
- `eps` a tolerance for the comparison of floating point numbers
- `verbose` print output or not
- `...` additional arguments

Examples

```r
x <- c(MM=298, MN=489, NN=213)
# Not run:
HW.test <- HWPerm(x,nperm=10000,verbose=TRUE) # uses default chi-square statistic
HW.test <- HWPerm(x,nperm=10000,verbose=TRUE,function(z)
HWChisq(z)$chisq,cc=0.5) # uses chi-square statistic with continuity correction.
HW.test <- HWPerm(x,nperm=10000,verbose=TRUE,function(y) HWLratio(y)$G2)
# uses likelihood ratio statistic.
HWPerm(x,nperm=10000,verbose=TRUE,function(y) 1-HWExact(y)$pval)
# uses exact test p-value
# # Permutation test for a marker on the X chromosome
# # rs5968922 <- c(A=392, B=212, AA=275, AB=296, BB=80)
HW.test <- HWPerm(rs5968922,nperm=10000,x.linked=TRUE,verbose=TRUE)
```

See Also

`HWChisq`, `HWExact`, `HLratio`
**Details**

This function approximates exact test probabilities for joint tests for HWE and equality of allele frequencies for variants with multiple alleles. For purely bi-allelic variant `HWPerm` can be used which allows for more statistics than just probabilities.

If argument `y` is not specified, gender is considered irrelevant, and `x` contains total genotype counts. If `x` and `y` are specified, `x` should contain male genotype counts and `y` female genotype counts. `x` and `y` can be vectors if the variant is bi-allelic, but are assumed lower triangular if there are more than two alleles. `x` is still a vector if there are multiple alleles but the variant is X-chromosomal. See the examples given below.

**Value**

- `pofthesample` probability of the observed sample
- `pseudodist` probabilities of simulated samples
- `pval` p-value

**Author(s)**

Jan Graffelman <jan.graffelman@upc.edu>

**References**


**See Also**

`HWPerm`

**Examples**

```r
# bi-allelic autosomal
#
x1 <- c(AA=298,AB=489, BB=213)
## Not run:
out <- HWPerm.mult(x1)
## End(Not run)
#
# bi-allelic X-chromosomal
#
x2.m <- c(A=39, B=21)
x2.f <- toTriangular(c(AA=28, AB=30, BB=8))
## Not run:
out <- HWPerm.mult(x2.m, x2.f)
```
## End(Not run)

#
# autosomal k alleles not accounting for gender
#

x3 <- c(AA=12,AB=19,AC=13,BB=7,BC=5,CC=0)
x3 <- toTriangular(x3)
## Not run:
out <- HWPerm.mult(x3)

## End(Not run)

#
# X-chromosomal k alleles
#

x4.m <- c(A=15,B=17,C=24)
x4.f <- toTriangular(c(AA=4,AB=2,AC=13,BB=6,BC=19,CC=4))
## Not run:
out <- HWPerm.mult(x4.m,x4.f)

## End(Not run)

#
# Autosomal k alleles accounting for gender
#

x5.m <- toTriangular(c(AA=12,AB=19,AC=13,BB=7,BC=5,CC=0))
x5.f <- toTriangular(c(AA=8,AB=12,AC=13,BB=8,BC=7,CC=0))
## Not run:
out <- HWPerm.mult(x5.m,x5.f)

## End(Not run)

#
# Autosomal STR with multipe alleles
#

data(NistSTRs)
A1 <- NistSTRs[,1]
A2 <- NistSTRs[,2]
GenotypeCounts <- AllelesToTriangular(A1,A2)
print(GenotypeCounts)
## Not run:
out <- HWPerm.mult(GenotypeCounts)

## End(Not run)
**HWPosterior**

*Calculation of posterior probabilities and Bayes factors for Hardy-Weinberg tests at X-chromosomal variants.*

**Description**

Function HWPosterior calculates posterior probabilities and Bayes factors for tests for Hardy-Weinberg equilibrium of autosomal and X-chromosomal variants.

**Usage**

```
HWPosterior(X, verbose = TRUE, prior.af = c(0.5,0.5), prior.gf =
             c(0.333,0.333,0.333), x.linked = FALSE, precision = 0.05)
```

**Arguments**

- `X`: A vector of genotype counts. The order c(A,B,AA,AB,BB) is assumed. Differently ordered vectors can be supplied but then elements must be labeled by their genotype.
- `verbose`: prints results if `verbose = TRUE`
- `prior.af`: Beta prior parameters for male and female allele frequencies
- `prior.gf`: Dirichlet prior parameters for female genotype frequencies
- `x.linked`: logical indicating whether the variant is autosomal or X-chromosomal
- `precision`: precision parameter for marginal likelihoods that require numeric integration

**Details**

For X-chromosomal variants, four possible models are considered, and the posterior probabilities and Bayes factors for each model are calculated.
For autosomal variants, ten possible scenarios are considered, and the posterior probabilities for all models are calculated.
In general, default Dirichlet priors are used for genotype frequencies, and beta prior are used for allele frequencies.

**Value**

For X-chromosomal variants, a matrix with posterior probabilities and Bayes factors will be produced. For autosomal variants, a vector of posterior probabilities is produced.

**Author(s)**

Xavi Puig <xavier.puig@upc.edu> and Jan Graffelman <jan.graffelman@upc.edu>

**References**

See Also

HWChisq, HWExact, HWExactStats

Examples

# An X-chromosomal example
#
x <- c(A=43, B=13, AA=26, AB=19, BB=3)
out <- HWPosterior(x, verbose=TRUE, x.linked=TRUE)
#
# An autosomal example
#
data(JPTsnps)
post.prob <- HWPosterior(JPTsnps[1,], x.linked=FALSE)

HWPower

Compute the power of a test for Hardy-Weinberg equilibrium.

Description

HWPower is a function that computes the power of a test for Hardy-Weinberg equilibrium.

Usage

HWPower(n = 100, nA = 100, pA = 0.5, y = c(AA=25, AB=50, BB=25),
alpha = 0.05, theta = 4, f = NULL, test = "exact",
alternative = "two.sided", pvaluetype = "selome", cc = 0.5)

Arguments

n The sample size
nA The minor allele count
pA The minor allele frequency
y A sample of genotype counts (AA, AB, BB)
alpha The significance level (0.05 by default)
theta The degree of disequilibrium (theta = 4 is equilibrium, theta > 4 is heterozygote excess, theta < 4 is heterozygote dearth)
f The inbreeding coefficient. Overrules theta if specified.
test The type of test for which power is to be computed. Can be "exact" (default) or "chisq" (chi-square)
alternative The nature of the alternative hypothesis ("two.sided" (default), "greater" or "less")
pvaluetype The type of p-value used in an exact test ("selome", "dost" or "midp")
cc Continuity correction parameter for the chi-square test (0.5 by default)
Details

HWPower uses the Levene-Haldane distribution (distribution of the number of heterzygotes given the minor allele count) for computing power.

HWPower can be used in three different ways. In principle, the power is calculated on the basis of the sample size ($n$) and the minor allele count ($n_A$). Alternatively, the user may specify sample size ($n$) and minor allele frequency ($p_A$). Finally, power can also be calculated directly from a sample of genotype counts. In that case the calculated power is the power for a sample of the given sample size and minor allele count. The three ways to use HWPower are illustrated in the example section.

Value

if test = "exact" the power of the exact test is computed for the given significance level and minor allele count.

if test = "chisq" the power of the chi-square test is computed for the given significance level and minor allele count.

Author(s)

Jan Graffelman (jan.graffelman@upc.edu)

References


See Also

HWExact

Examples

```r
pw.chisq <- HWPower(n=100,nA=100,alpha=0.05,test="chisq",theta=16)
print(pw.chisq)
pw.exact <- HWPower(n=100,nA=100,alpha=0.05,test="exact",theta=16,pvaluetype="selome")
print(pw.exact)
pw.exact <- HWPower(n=100,nA=100)
print(pw.exact)
pw.exact <- HWPower(n=100,pA=0.5)
print(pw.exact)
pw.exact <- HWPower(y=c(AA=25,AB=50,BB=25))
print(pw.exact)
```
**HWQqplot**

*A Q-Q plot for Hardy-Weinberg equilibrium*

**Description**

HWQqplot creates a Q-Q plot for the p-values obtained in an Exact test for Hardy-Weinberg equilibrium. Empirical p-values are plotted against multiple simulated quantiles of the theoretical p-value distribution.

**Usage**

```r
HWQqplot(X, nsim = 100, fit = "curve", logplot = FALSE,
         main = "Q-Q plot for HWE", mm = NULL, pvaluetype = "selome", ...)
```

**Arguments**

- **X**: Data matrix with genotype counts, one row for each sample, 3 columns
- **nsim**: Number of samples drawn from the null distribution (100 by default)
- **fit**: If fit is set to "line" straight lines will be fitted to the simulated samples, if set to "curve", ascending curves will be shown.
- **logplot**: If logplot is set to true, then the log10 of the p-values will be used in the plot. If not, untransformed p-values will be used.
- **main**: Title for the plot
- **mm**: Maximal value for x and y axis in the plot
- **pvaluetype**: Type of p-value to be used in an exact test. Can be "selome" (default), "midp" or "dost".
- **...**: Any additional arguments for the `plot` instruction

**Details**

HWQqplot constructs a Q-Q plot of the p-values of an exact test for Hardy-Weinberg equilibrium. Under the null, this p-value is not uniform. HWQqplot samples from the theoretical null distribution, taking into account that markers may vary in allele frequency and in sample size (due to missing values). For each simulated sample a grey curve or line is shown. A green reference line with intercept 0 and slope 1 is also shown in the plot.

**Value**

`NULL`

**Author(s)**

Jan Graffelman <jan.graffelman@upc.edu>
References


See Also

HWTernaryPlot HWExact qqplot

Examples

## Not run:
set.seed(1234)
n <- 200 # sample size
m <- 100 # number of markers
X <- HWData(n,m)$X
HWQqplot(X,logplot=TRUE,pvaluetype="selome",main="Q-Q Plot for HWE")

## End(Not run)

Description

HWTernaryPlot is a routine that draws a ternary plot for three-way genotypic compositions (AA,AB,BB), and represents the acceptance region for different tests for Hardy-Weinberg equilibrium (HWE) in the plot. This allows for graphical testing of a large set of markers (e.g. SNPs) for HWE. The (non) significance of the test for HWE can be inferred from the position of the marker in the ternary plot. Different statistical tests for HWE can be done graphically with this routine: the ordinary chi-square test, the chi-square test with continuity correction and the Haldane’s exact test.

Usage

HWTernaryPlot(X, n = NA, addmarkers = TRUE, newframe = TRUE, hwcurve = TRUE, vbounds = TRUE, mafbounds = FALSE, mafvalue = 0.05, axis = 0, region = 1, vertexlab = colnames(X), alpha = 0.05, vertex.cex = 1, pch = 19, cc = 0.5, markercol = "black", markerbgcol = "black", cex = 0.75, axislab = "", verbose = FALSE, markerlab = NULL, markerpos = NULL, mcex = 1, connect = FALSE, curvecols = rep("black",5), signifcolour = TRUE, curtyp = "solid", ssf = "max", pvaluetype = "selome", ...)
be used for painting acceptance regions. If not supplied \( n \) is computed from the data in \( X \).

\( n \)  
the samples size (for a complete composition with no missing data).

\( \text{addmarkers} \)  
represent markers by dots in the triangle (\text{addmarkers}=\text{TRUE}) or not (\text{addmarkers}=\text{FALSE}).

\( \text{newframe} \)  
allows for plotting additional markers in an already existing ternary plot. Overplotting is achieved by setting \text{newframe} to \text{FALSE}. Setting \text{newframe} = \text{TRUE} (default) will create a new ternary plot.

\( \text{hwcurve} \)  
draw the HW parabola in the plot (\text{hwcurve}=\text{TRUE}) or not (\text{hwcurve}=\text{FALSE}).

\( \text{vbounds} \)  
indicate the area corresponding to expected counts > 5 (\text{vbounds}=\text{TRUE}) or not (\text{vbounds}=\text{FALSE}).

\( \text{mafvalue} \)  
a critical value for the minor allele frequency (MAF).

\( \text{axis} \)  
draw a vertex axis  
0 = no axis is drawn  
1 = draw the AA axis  
2 = draw the AB axis  
3 = draw the BB axis

\( \text{region} \)  
the type of acceptance region to be delimited in the triangle  
0 = no acceptance region is drawn  
1 = draw the acceptance region corresponding to a Chi-square test  
2 = draw the acceptance region corresponding to a Chi-square test with continuity correction  
3 = draw the acceptance region corresponding to a Chi-square test with continuity correction for \( D > 0 \)  
4 = draw the acceptance region corresponding to a Chi-square test with continuity correction for \( D < 0 \)  
5 = draw the acceptance regions for all preceding tests simultaneously  
6 = draw the acceptance region corresponding to a Chi-square test with continuity correction with the upper limit for \( D > 0 \) and the lower limit for \( D < 0 \)  
7 = draw the acceptance region corresponding to a two-sided exact test

\( \text{vertexlab} \)  
labels for the three vertices of the triangle

\( \text{alpha} \)  
significance level (0.05 by default)

\( \text{vertex.cex} \)  
character expansion factor for the labels of the vertices of the triangle.

\( \text{pch} \)  
the plotting character used to represent the markers.

\( \text{cc} \)  
value for the continuity correction parameter (0.5 by default).

\( \text{markercol} \)  
vector with colours for the marker points in the triangle.

\( \text{markerbgcol} \)  
vector with background colours for the marker points in the triangle.

\( \text{cex} \)  
expansion factor for the marker points in the triangle.

\( \text{axislab} \)  
a label to be put under the horizontal axis.
verbose
labels for the markers in the triangle.
positions for the marker labels in the triangle (1,2,3 or 4).
character expansion factor for the labels of the markers in the ternary plot.
connect the represented markers by a line in the ternary plot.
a vector with four colour specifications for the different curves that can be used
to delimit the HW acceptance region. E.g. curvecols=c("red","green","blue","black","purple")
will paint the Hardy-Weinberg curve red, the limits of the acceptance region for
an ordinary chi-square test for HWE green, the limits of the acceptance region
for a chi-square test with continuity correction when D > 0 blue and the limits
of the acceptance region for a chi-square test with continuity correction when D
< 0 black, and the limits of the exact acceptance region purple.
colour the marker points automatically according to the result of a signifance
test (green markers non-significant, red markers significant). signifcolour
only takes effect if region is set to 1, 2 or 7.
style of the drawn curves ("dashed","solid","dotted",...)
sample size function ("max","min","mean","median",...). Indicates how
the sample size for drawing acceptance regions is determined from the matrix
counts.
method to compute p-values in an exact test ("dost" or "selome")
other arguments passed on to the plot function (e.g. main for a main title).

Details
HWternaryPlot automatically colours significant markers in red, and non-significant markers in
green if region is set to 1, 2 or 7.

Value
minimum allele frequency above which testing for HWE is appropriate (ex-
expected counts exceeding 5).
maximum allele frequency below which testing for HWE is appropriate.
number of markers in the appropriate range.
percentage of markers in the appropriate.
number of significant markers (only if region equals 1,2 or 7.)

Author(s)
Jan Graffelman <jan.graffelman@upc.edu>

References
the Ternary Plot. *Human Heredity* 65(2):77-84.
of Statistical Software* 64(3): 1-23. [https://www.jstatsoft.org/v64/i03/](https://www.jstatsoft.org/v64/i03/).
See Also

HWChisq

Examples

n <- 100 # sample size
m <- 100 # number of markers

X <- HWData(n,m)
HWternaryPlot(X,100,region=1,hwcurve=TRUE,vbounds=FALSE,vertex.cex=2)

HWTriExact

Exact test for Hardy-Weinberg equilibrium and equality of allele frequencies for tri-allelic variants.

Description

Function HWTriExact does a standard exact test for Hardy-Weinberg equilibrium of a tri-allelic variant, and also does joint exact tests for equilibrium and equality of allele frequencies if the genotype counts are given separately for both sexes

Usage

HWTriExact(x, y = NULL, eps = 1e-10, nperm = 17000, verbose = TRUE)

Arguments

x vector with 6 genotype counts (AA,AB,AC,BB,BC,CC)
y vector with 6 or 3 genotype counts (AA,AB,AC,BB,BC,CC) or (A,B,C)
eps a tolerance that can be set for comparing exact probabilities
nperm number of permutations (only relevant for autosomal stratified by gender)
verbose print output or not

Details

If only x is specified, an exact test for an autosomal variant with three alleles will be performed.
If both x and y are supplied as vectors with 6 elements, a permutation test for HWE and equality of allele frequencies (EAF) for an autosomal variant is performed, using nperm permutations. The distribution of the probabilities is returned in pseudodist. The computational cost of a completed enumeration algorithm can be prohibitive in this case.
If x is supplied as a length 6 vector, and y as a length 3 vector, the variant is assumed to be X-chromosomal, x containing female genotype counts and y containing male genotype counts. In this case a joint exact test for HWE and EAF for an X-chromosomal tri-allelic variant is executed.
See the examples in the example section below.
**Value**

- `pval`: The p-value of the sample
- `pseudodist`: Distribution of probabilities obtained by simulation
- `pofthesample`: The probability of the observed sample

**Author(s)**

Jan Graffelman <jan.graffelman@upc.edu>

**References**


**See Also**

`HWPerm.mult`

**Examples**

```r
# Autosomal tri-allelic (not accounting for gender)
#
x <- c(AA=20,AB=31,AC=26,BB=15,BC=12,CC=0)
## Not run: out <- HWTriExact(x)
#
# Autosomal tri-allelic accounting for gender
#
males <- c(A=1,B=21,C=34)
females <- c(AA=0,AB=1,AC=0,BB=8,BC=24,CC=15)
## Not run: out <- HWTriExact(females,males)
#
# X-chromosomal tri-allelic accounting for gender
#
males <- c(A=1,B=21,C=34)
females <- c(AA=0,AB=1,AC=0,BB=8,BC=24,CC=15)
## Not run: out <- HWTriExact(females,males)
```
**ifisherz**

**Inverse Fisher z transformation**

**Description**

Calculates the inverse of Fisher’s z transformation

**Usage**

`ifisherz(y)`

**Arguments**

- `y` a real number

**Value**

a correlation coefficient in the range (-1,1)

**Author(s)**

Jan Graffelman (jan.graffelman@upc.edu)

**See Also**

`cor`

**Examples**

```r
r <- 0.5
print(ifisherz(fisherz(r)))
```

**JPTmultiallelicsChr7**

Multi-allelic autosomal variants of the Japanese population of the 1000 genomes project

**Description**

JPTtriallelicsChrX contains three selected multi-allelic variants on chromosome 7 from the Japanese sample of the 1000 genomes project.

**Usage**

`data("JPTmultiallelicsChr7")`
Format

List object with fields m4,f4; m5,f5; m6,f6;

Details

The list object contains male and female genotype counts for 3 multi-allelic variants on chromosome 7 of the JPT sample of the 1000 genomes project.

Source

The The 1000 genomes project.

References


Examples

data(JPTmultiallelicsChr7)
str(JPTmultiallelicsChr7)

---

JPTmultiallelicsChrX  Multi-allelic X-chromosomal variants of the Japanese population of the 1000 genomes project

Description

JPTtriallelicsChrX contains four selected multi-allelic variants on the X chromosome from the Japanese sample of the 1000 genomes project.

Usage

data("JPTmultiallelicsChrX")

Format

List object with fields m4,f4; m5,f5; m6,f6; m7,f7

Details

The list object contains male and female genotype counts for four multi-allelic variants of the JPT sample of the 1000 genomes project.

Source

The The 1000 genomes project.
References


Examples

data(JPTmultiallelicsChrX)
m4 <- JPTmultiallelicsChrX$m4
f4 <- JPTmultiallelicsChrX$f4

---

**JPTsnps**  
*Bi-allelic SNPs from a Japanese population*

Description

JPTsnps contains genotype counts for the two sexes of ten single nucleotide polymorphisms of the Japanese (JPT) sample of the 1000 Genomes project.

Usage

data("JPTsnps")

Format

data frame

Source

The The 1000 genomes project.

References


Examples

data(JPTsnps)
Description

JPTtriallelics contains six selected tri-allelic variants on chromosome 7 from the Japanese sample of the 1000 genomes project.

Usage

data("JPTtriallelicsChr7")

Format

A data frame with 6 observations on the following 14 variables.

id  RS identifier
pos position in base pairs
mAa number of AA males
mAb number of AB males
mAc number of AC males
mBb number of BB males
mBc number of BC males
mCc number of CC males
fAa number of AA females
fAb number of AB females
fAc number of AC females
fBb number of BB females
fBc number of BC females
fCc number of CC females

Source

The The 1000 genomes project.

References


Examples

data(JPTtriallelicsChr7)
str(JPTtriallelicsChr7)
**Description**

`JPTtriallelicsChrX` contains five selected tri-allelic variants on the X chromosome from the Japanese sample of the 1000 genomes project.

**Usage**

```r
data("JPTtriallelicsChrX")
```

**Format**

A data frame with 5 observations on the following 12 variables.

- **id**: Identifier of the polymorphism
- **pos**: Position of the polymorphism in base pairs
- **chr**: Chromosome
- **A**: Number of males with A genotype
- **B**: Number of males with B genotype
- **C**: Number of males with C genotype
- **AA**: Number of AA females
- **AB**: Number of AB females
- **AC**: Number of AC females
- **BB**: Number of BB females
- **BC**: Number of BC females
- **CC**: Number of CC females

**Source**

The 1000 genomes project.

**References**


**Examples**

```r
data(JPTtriallelicsChrX)
str(JPTtriallelicsChrX)
```
mac

Compute the minor allele count.

Description

mac computes the smallest allele count for a given vector of genotype counts.

Usage

mac(X)

Arguments

X  a vector or matrix with genotype counts (AA, AB, BB)

Value

a vector of the minor allele counts

Author(s)

Jan Graffelman (jan.graffelman@upc.edu)

See Also

maf

Examples

X <- as.vector(rmultinom(1,100,c(0.5,0.4,0.1)))
names(X) <- c("AA","AB","BB")
print(X)
print(mac(X))

maf

Function to compute minor allele frequencies

Description

Function maf computes the minor allele frequency for a matrix or vector of compositions.

Usage

maf(x)
MakeCounts

Create genotype counts from bi-allelic marker data

Description

MakeCounts creates a matrix of genotype counts, with one row for each bi-allelic marker, containing 4 columns with the counts AA, AB, BB and NA (missings) respectively.

Usage

MakeCounts(X, alleles, pos1 = 1, pos2 = 3, coding = c(AA=0,AB=1,BB=2), sep = "")

Arguments

X  A matrix or dataframe with bi-allelic genotyping information, markers in columns, individuals in rows
alleles  a vector of alleles for each marker (e.g. c("A/T","A/G",...)). Only relevant if X is a matrix with text entries.
pos1  position of the first allele in the allele string (1 by default)
pos2  position of the second allele in the allele string (3 by default)
coding  indicates how homozygotes and heterozygote are coded as numbers. Only relevant if X is a matrix with numeric entries.
sep  allele separator character for genotype data in text format ("" for AA; "/" for "A/A")

Examples

X <- as.vector(rmultinom(1,100,c(0.5,0.4,0.1)))
X <- X/sum(X)
print(X)
print(maf(X))
Details

MakeCounts is thought for bi-allelic marker data only. Missings are should be coded by NA. It produces the right input for HW3TernaryPlot.

Heterozygotes may be coded in the data as "AB" or "BA". Both entries will be counted as a heterozygote.

Value

A matrix of 4 columns

Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

See Also

HW3TernaryPlot

Examples

SNP1 <- c("GG", "GG", "GG", "GG", "GG", "GG", "GG")
SNP2 <- c("CG", "CG", "CC", "GG", "GG", "CG", "CG")
SNP3 <- c("AA", "AA", "AG", "AA", "AG", "AA", "AA")
SNP4 <- c("GG", "GG", "GG", "GG", "GG", "GG", "GG")
SNP5 <- c("CC", "CC", "CC", "CC", "CC", "CT", "CT")
X <- cbind(SNP1, SNP2, SNP3, SNP4, SNP5)
print(Y)
W <- matrix(sample(c(0,1,2,NA),100,replace=TRUE),ncol=5)
Z <- MakeCounts(W, coding=c(0,1,2))
Details

If \( x \) is a factor, it will be coerced to a factor with levels AA, AB and BB. Important detail: the produced factors will have only those levels that are observed in the data. E.g., if genotyping results only consist of \((0,1)\), then the resulting factor will not have level BB (which would be an empty category).

Value

A factor variable

Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

See Also

MakeCounts

Examples

```r
y <- c(1,1,0,0,2,2)
data.frame(y,MakeFactor(y))

y <- c(2,2,3,3,1,1)
data.frame(y,MakeFactor(y,coding=c(1,2,3)))

data(Markers)
data.frame(Markers[,1],MakeFactor(Markers[,1],coding=c(1,2,3)))
```

**Markers**

<table>
<thead>
<tr>
<th>SNP data and intensities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Markers</td>
</tr>
</tbody>
</table>

Description

The dataframe contains the genotypes of 3 SNPs and two allele intensities of 146 individuals. The first column is a GT polymorphism that has missing values for several individuals. The second and third column (iG and iG) are the allele intensities of this polymorphism. Column 4 and 5 are covariate SNPs (an AC and an AG polymorphism) that have no missing values.

Usage

```r
data(Markers)
```

Format

A data frame containing 146 rows and 5 columns
References


Mourant

Genotype frequencies for blood group locus MN

Description

The dataframe contains the genotype frequencies MM, MN and NN for the MN blood group locus for 216 populations. The data are taken from table 2.5 in Mourant et al., using only entries with a sample size of at least 500.

Usage

data(Mourant)

Format

A data frame containing 216 observations.

Source

Mourant et al, Table 2.5

References


n.alleles

Number of alleles

Description

Function n.alleles determines the number of alleles in a named genotype vector.

Usage

n.alleles(x, ...)

Arguments

x A named genotype vector (e.g. c(AA=10,AB=20,BB=5))

... extra arguments that are passed on to alleles
Value
  integer

Author(s)
  Jan Graffelman (jan.graffelman@upc.edu)

See Also
  alleles

Examples
  x <- c(AA=25, AB=50, BB=25)
  k <- n.alleles(x)
  print(k)

---

NistSTRs  NIST autosomal STR data

Description
  NistSTRs contains 29 autosomal microsatellites (STRs) of individuals of Caucasian ancestry. The two alleles of an individual are separated into two columns for each STR.

Usage
  data("NistSTRs")

Format
  A data frame with 361 observations on the following 58 variables.

  ‘CSF1PO-1’  First allele of CSF1PO
  ‘CSF1PO-2’  Second allele of CSF1PO
  ‘D10S1248-1’  First allele of D10S1248
  ‘D10S1248-2’  Second allele of D10S1248
  ‘D12S391-1’  First allele of D12S391
  ‘D12S391-2’  Second allele of D12S391
  ‘D13S317-1’  First allele of D13S317
  ‘D13S317-2’  Second allele of D13S317
  ‘D16S539-1’  First allele of D16S539
  ‘D16S539-2’  Second allele of D16S539
  ‘D18S51-1’  First allele of D18S51
  ‘D18S51-2’  Second allele of D18S51
D19S433-1’ First allele of D19S433
D19S433-2’ Second allele of D19S433
D1S1656-1’ First allele of D1S1656
D1S1656-2’ Second allele of D1S1656
D21S11-1’ First allele of D21S11
D21S11-2’ Second allele of D21S11
D22S1045-1’ First allele of D22S1045
D22S1045-2’ Second allele of D22S1045
D2S1338-1’ First allele of D2S1338
D2S1338-2’ Second allele of D2S1338
D2S441-1’ First allele of D2S441
D2S441-2’ Second allele of D2S441
D3S1358-1’ First allele of D3S1358
D3S1358-2’ Second allele of D3S1358
D5S818-1’ First allele of D5S818
D5S818-2’ Second allele of D5S818
D6S1043-1’ First allele of D6S1043
D6S1043-2’ Second allele of D6S1043
D7S820-1’ First allele of D7S820
D7S820-2’ Second allele of D7S820
D8S1179-1’ First allele of D8S1179
D8S1179-2’ Second allele of D8S1179
F13A01-1’ First allele of F13A01
F13A01-2’ Second allele of F13A01
F13B-1’ First allele of F13B
F13B-2’ Second allele of F13B
FESFPS-1’ First allele of FESFPS
FESFPS-2’ Second allele of FESFPS
FGA-1’ First allele of FGA
FGA-2’ Second allele of FGA
LPL-1’ First allele of LPL
LPL-2’ Second allele of LPL
Penta_C-1’ First allele of Penta_C
Penta_C-2’ Second allele of Penta_C
Penta_D-1’ First allele of Penta_D
Penta_D-2’ Second allele of Penta_D
Penta_E-1’ First allele of Penta_E
‘Penta_E-2’ Second allele of Penta_E
‘SE33-1’ First allele of SE33
‘SE33-2’ Second allele of SE33
‘TH01-1’ First allele of TH01
‘TH01-2’ Second allele of TH01
‘TPOX-1’ First allele of TPOX
‘TPOX-2’ Second allele of TPOX
‘vWA-1’ First allele of vWA
‘vWA-2’ Second allele of vWA

Source
http://strbase.nist.gov

References

Examples
data(NistSTRs)

<table>
<thead>
<tr>
<th>recode</th>
<th>Recode genotype information</th>
</tr>
</thead>
</table>

Description
function recode recodes bi-allelic genetic marker information expressed as strings (e.g. "AA", "AB", "BB") into numerical form.

Usage
recode(X, alleles, values = c(0, 1, 2), pos1 = 1, pos2 = 3, minor = FALSE, verbose = FALSE)

Arguments
X A matrix or dataframe of bi-allelic markers, individuals in rows, markers in columns
alleles a vector with the alleles for each marker (e.g. c("A/T", "A/G", etc))
values a vector of numerical values for AA, AB and BB, ((0,1,2) by default).
pos1 position of the first allele in the allele string (1 by default).
pos2 position of the second allele in the allele string (3 by default).
minor coding is according to the number of copies of the minor allele. If `minor = TRUE`, the value of 2 reflects two copies of the minor allele, and the value 0 reflects no copies of the minor allele.

verbose: print progress on the conversion or not.

Details

recode is written for bi-allelic marker data only. Heterozygotes may be coded both as AB or BA. By default, the second allele specified (e.g. "T" in "A/T") is counted in the recoding, and homozygotes AA are coded as 0 and homozygotes TT as 2.

Value

A numerical matrix, individuals in rows, markers in columns

Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

See Also

MakeCounts

Examples

```r
SNP1 <- c("GG", "GG", "GG", "GG", "GG", "GG", "GG")
SNP2 <- c("CG", "GG", "CC", "GG", "CG", "CG", "CG")
SNP3 <- c("AA", "AA", "AA", "AG", "AA", "AA", "AA")
SNP4 <- c("GG", "GG", "GG", "GG", "GG", "GG", "GG")
SNP5 <- c("CC", "CC", "CC", "CC", "CC", "CT", "CT", "CT")
X <- cbind(SNP1, SNP2, SNP3, SNP4, SNP5)
print(Y)
```

---

### strsort

#### Description

Function `strsort` collapses all tokens of a vector of strings in a single string with sorted tokens.

#### Usage

`strsort(s)`

#### Arguments

- `s`: a vector of character strings

---

Function `strsort` collapses all tokens of a vector of strings in a single string with sorted tokens.
**Value**

a string

**Author(s)**

Jan Graffelman <jan.graffelman@upc.edu>

**See Also**

alleles

**Examples**

```r
x <- c("AA", "AB", "BB", "AC", "CC")
print(strsort(x))
```

---

**ThetatoF**

*Convert theta to an inbreeding coefficient*

**Description**

Function ThetatoF converts disequilibrium measure theta to an inbreeding coefficient.

**Usage**

```r
ThetatoF(p, theta = 4)
```

**Arguments**

- `p`:
  - the allele frequency
- `theta`:
  - the disequilibrium parameter

**Value**

a real number

**Author(s)**

Jan Graffelman (jan.graffelman@upc.edu)

**References**


**See Also**

HWf
toTriangular

Examples

f <- ThetatoF(0.5, 4)

toTriangular

Convert a vector of genotype counts to triangular format

Description

Function toTriangular converts a named vector of genotype counts into a triangular matrix format, with homozygotes on the diagonal and heterozygotes below the diagonal.

Usage

toTriangular(x)

Arguments

x

A vector of genotype counts

Value

a matrix

Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

Examples

x <- c(AA=20, AB=52, AC=34, BB=17, BC=51, CC=26)
print(x)
X <- toTriangular(x)
print(X)

TSIXTriAllelics

Tri-allelic polymorphisms on the X chromosome of the TSI population

Description

This dataframe contains genotype counts for six three-allelic polymorphisms (A,B,C) on chromosome X of a sample of individuals from the TSI population (Tuscany, Italy) of the 1,000 genomes project.

Usage

data(TSIXTriAllelics)
Format

A data frame with 6 observations on the following 10 variables.

ID  Identifier of the polymorphism
A  Male A genotype count
B  Male B genotype count
C  Male C genotype count
AA  Female AA genotype count
AB  Female AB genotype count
AC  Female AC genotype count
BB  Female BB genotype count
BC  Female BC genotype count
CC  Female CC genotype count

Source

Data taken from the 1,000 genomes project at www.internationalgenome.org

References


Examples

data(TSIXTriAllelics)

UniqueGenotypeCounts

Extract unique genotypic compositions from a matrix

Description

Function UniqueGenotypeCounts creates a matrix containing only the unique rows in the given matrix, together with their frequency of occurrence

Usage

UniqueGenotypeCounts(X, verbose = TRUE)

Arguments

X  A n by 3 matrix with genotypic counts (AA,AB,BB)
verbose  If TRUE then print some statistics
vaf

Value

A matrix with 4 columns, AA, AB, BB, and frequency of occurrence

Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

See Also

GenerateSamples

Examples

set.seed(123)
X <- HWData(n=100,nm=100)
print(nrow(X))
Y <- UniqueGenotypeCounts(X)
print(nrow(Y))
print(sum(Y$w))

vaf

Computes the sample variance of the allele frequency for a biallelic marker.

Description

Function vaf computes the sample variance of the allele frequencies of a single sample or a matrix of samples.

Usage

vaf(X, hw = FALSE)

Arguments

X vector or matrix with genotype counts (AA,AB,BB)

hw assume Hardy-Weinberg proportions (hw=TRUE) or not (hw=FALSE)

Details

For biallelic markers the variance of the minor allele frequency equals the variance of the major allele frequency.

Value

a numeric vector of variances.
Author(s)

Jan Graffelman <jan.graffelman@upc.edu>

References


See Also

af, maf

Examples

```r
x <- c(MM=298, MN=489, NN=213)
pA <- af(x)
vA <- vaf(x)
cat("allele frequency:", pA, "\n")
cat("sample variance allele frequency:", vA, "\n")
```
Index

* ~misc
  * genlabels, 17
* aplot
  * HWAlrPlot, 22
  * HWClrPlot, 29
  * HWGenotypePlot, 40
  * HWIlrPlot, 42
  * HWAqplot, 58
  * HWternaryPlot, 59
* array
  * fold, 15
* datagen
  * GenerateSamples, 16
  * HWData, 32
* datasets
  * Alzheimer, 9
  * HapMapCHBChr1, 17
  * JPTmultiallelicsChr7, 64
  * JPTmultiallelicsChrX, 65
  * JPTsnpS, 66
  * JPTtriaIlelicsChr7, 67
  * JPTttriallelicsChrX, 68
  * Markers, 72
  * Mourant, 73
  * NistSTRs, 74
  * TSIXTriAllelics, 79
* distribution
  * dgraffelmanweir.bi, 10
  * dIeIene, 11
  * dIeIene.bi, 12
  * HWIindley, 43
* htest
  * AFtest, 5
  * EAExaCt, 13
  * HWAlltests, 20
  * HWChisq, 23
  * HWChisqMat, 25
  * HWChisqStats, 26
  * HWEaCt, 33
  * HWEaCtMat, 35
  * HWEaCtPrevious, 36
  * HWEaCtStats, 37
  * HWLaCt, 45
  * HWLRtest, 46
  * HWNnetwork, 49
  * HWPerm, 51
  * HWPosterior, 55
  * HWPower, 56
* manip
  * fold, 15
* misc
  * af, 5
  * agcounts, 6
  * alleles, 7
  * AllelesToTriangular, 8
  * fisherz, 14
  * HWABO, 18
  * HWaIC, 19
  * HWAlr, 21
  * HWClr, 27
  * HWClr, 28
  * HWCondProbAB, 30
  * HWD, 31
  * HWf, 39
  * HWIlr, 41
  * HWLRAII Tests, 44
  * HWMissing, 47
  * HWPermmult, 52
  * HWTriExact, 62
  * ifisherz, 64
  * mac, 69
  * maf, 69
  * MakeCounts, 70
  * MakeFactor, 71
  * n.alleles, 73
  * recode, 76
  * strsort, 77
  * ThetatoF, 78
INDEX

index toTriangular, 79
    UniqueGenotypeCounts, 80
vaf, 81

* package
    HardyWeinberg-package, 3

af, 5, 18, 82
AFtest, 5
agcounts, 6
alleles, 7, 74, 78
AllelesToTriangular, 8
Alzheimer, 9
cor, 15, 64
dgRaffelmanweir.bi, 10
dlevene, 11, 12
dlevene.bi, 12
EAFExact, 13
fisher.test, 13
fisherz, 14
fold, 15
GenerateSamples, 16, 81
genlabels, 17
HapMapCHBChr1, 17
HardyWeinberg (HardyWeinberg-package), 3
HardyWeinberg-package, 3
HWABO, 18
HWAIC, 19, 47
HWAllTests, 20
HWAlr, 21, 27, 28, 41
HWAlrPlot, 22, 29, 42
HWChisq, 6, 17, 21, 23, 25, 26, 31, 34, 37, 39, 45, 49, 52, 56, 62
HWChisqMat, 25
HWChisqStats, 24, 26
HWClo, 27
HWClr, 21, 27, 28, 41
HWClrPlot, 22, 29, 42
HWCondProbAB, 30
HWD, 31
HWData, 32
HWExact, 6, 10–12, 21, 30, 33, 35, 38, 50, 52, 56, 57, 59
HWExactMat, 35
HWExactPrevious, 36
HWExactStats, 10, 34, 37, 50, 56
Hwf, 31, 39, 78
HWGenotypePlot, 40
HWIlr, 21, 27, 28, 41
HWIlrPlot, 22, 29, 42
HWLindley, 43
HWLRAIITests, 44
HWWratio, 21, 24, 34, 37, 45, 52
HWLRtest, 20, 44, 46
HWWMissing, 47
HWNetwork, 49
HWPerm, 51, 53
HWPerm.multi, 52, 63
HWPosterior, 43, 54
HWPower, 56
HWQplot, 58
HWternaryPlot, 33, 41, 59, 59, 71
HWTriExact, 62
ifisherz, 64
JTmultiallallelicsChr7, 64
JTmultialallelicsChrX, 65
JPTsnp, 66
JTtriallallelicsChr7, 67
JTtriallallelicsChrX, 68
lower.tri, 15
mac, 7, 69
maf, 5, 69, 69, 82
MakeCounts, 70, 72, 77
MakeFactor, 71
Markers, 72
Mourant, 73
n.alleles, 8, 73
NistSTRs, 74
qqplot, 59
recode, 76
strsort, 77
ThetatoF, 78
toTriangular, 9, 79
TSIXTriAllelics, 79
UniqueGenotypeCounts, 80
upper.tri, 15
vaf, 81