Package: hapsim (via r-universe)

June 10, 2026

Title Haplotype Data Simulation

Version 0.31

Date 2017-06-05

Author Giovanni Montana

Maintainer Apostolos Dimitromanolakis <apostolis@live.ca>

Description Package for haplotype-based genotype simulations. Haplotypes are generated such that their allele frequencies and linkage disequilibrium coefficients match those estimated from an input data set.

Depends MASS

License GPL (>= 2)

NeedsCompilation yes

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

Date/Publication 2017-06-07 20:40:03 UTC

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

RemoteRef HEAD

RemoteSha a4fd4ef5e73ec3fab1d1a13f405b5a2f9f0b0120

Contents

ACEdata *ACE data set*

Description

ACE (angiotensin I converting enzyme) data set

Usage

```
data(ACEdata)
```

Format

A data set with 22 haplotypes and 52 SNPs.

References

Montana, G. HapSim: a simulation tool for generating haplotype data with pre-specified allele frequencies and LD coefficients. 2005.

allelefreqs *Estimates allele frequencies*

Description

Estimates allele frequencies from a binary matrix

Usage

```
allelefreqs(dat)
```

Arguments

dat	A binary matrix, rows are haplotypes and columns are binary markers

Value

A list containing:

freqs	Vector of allele "0" frequencies
all.polym	If TRUE, all loci are polymorphic
non.polym	Vector of non-polymorphic loci, if any

Author(s)

Giovanni Montana

References

Montana, G. HapSim: a simulation tool for generating haplotype data with pre-specified allele frequencies and LD coefficients. 2005.

Examples

```
data(ACEdata)
x <- allelefreqs(ACEdata)
hist(x$freqs)
```

`divlocus` *Diversity score*

Description

Compute a measure of genetic diversity at each locus

Usage

```
divlocus(dat)
```

Arguments

`dat`	A binary matrix, rows are haplotypes and columns are binary markers

Details

This function implements a measure of diversity for a locus j as in Clayton (2002). If $z_i j$ represents the allele j of haplotype i, for $i = 1, ..., N$ and assuming that alleles are coded as 0 and 1, the diversity measure can be written as

$$D_j = 2 * N(\sum_{i=1}^{N} z_{ij}^2 - (\sum_{i=1}^{N} z_{ij})^2)$$

Value

A vector containing the diversity measure for all markers

Author(s)

Giovanni Montana

References

D. Clayton. Choosing a set of haplotype tagging SNPs from a larger set of diallelic loci. 2002. www-gene.cimr.cam.ac.uk/clayton/software/stata/htSNP/htsnp.pdf

Examples

```
data(ACEdata)
divlocus(ACEdata)
```

`haplodata` *Haplotype object creator*

Description

Creates an haplotype data object needed for simulating haplotypes with `haplosim`. This object also contains some summary statistics about the real data.

Usage

```
haplodata(dat)
```

Arguments

`dat`	A binary matrix, rows are haplotypes and columns are binary markers

Value

A list containing:

`freqs`	Allele frequencies
`cor`	Correlation matrix (LD coefficients)
`div`	Locus-specific diversity measure
`cov`	Covariance matrix for the normal distribution

Author(s)

Giovanni Montana

References

Montana, G. HapSim: a simulation tool for generating haplotype data with pre-specified allele frequencies and LD coefficients. 2005.

See Also

See also `haplosim`

Examples

```
data(ACEdata)

# creates the haplotype object
x <- haplodata(ACEdata)

# simulates 100 random haplotypes
y <- haplosim(100, x)
```

---

haplofreqs *Haplotype frequencies*

Description

Compute haplotype frequencies

Usage

```
haplofreqs(dat, firstl, lastl)
```

Arguments

dat	A binary matrix, rows are haplotypes and columns are binary markers
firstl	Position of the first locus
lastl	Position of the last locus

Value

A vector of haplotype frequencies

Author(s)

Giovanni Montana

References

Montana, G. HapSim: a simulation tool for generating haplotype data with pre-specified allele frequencies and LD coefficients. 2005.

Examples

```
data(ACEdata)
freqs <- haplofreqs(ACEdata, 17, 22)
```

`haplosim` *Haplotype data simulator*

Description

Generates a random sample of haplotypes, given an haplotype object created from a data set

Usage

```
haplosim(n, hap, which.snp = NULL, seed = NULL, force.polym = TRUE, summary = TRUE)
```

Arguments

`n`	Number of haplotypes to generate
`hap`	Haplotype object created with `haplodata`
`which.snp`	A vector specifying which SNPs to include
`seed`	Seed for the random number generator
`force.polym`	if TRUE, all loci are polymorphic
`summary`	if TRUE, additional summary statistics are returned

Value

A list containing:

`data`	Simulated sample
`freqs`	Allele frequency vector
`cor`	Correlation matrix
`div`	Locus-specific diversity scores
`mse.freqs`	MSE of allele frequencies
`mse.cor`	MSE of correlations

Author(s)

Giovanni Montana

References

Montana, G. HapSim: a simulation tool for generating haplotype data with pre-specified allele frequencies and LD coefficients. 2005.

See Also

See also `haplodata`

Examples

```
#
# Example 1
#

data(ACEdata)

# create the haplotype object
x <- haplodata(ACEdata)

# simulates a first sample of 100 haplotypes using all markers
y1 <- haplosim(100, x)

# compares allele frequencies in real and simulated samples
plot(x$freqs, y1$freqs, title=paste("MSE:",y1$mse.freqs)); abline(a=0, b=1)

# compares LD coefficients in real and simulated samples
ldplot(mergemats(x$cor, y1$cor), ld.type='r')

# simulates a second sample of 1000 haplotypes using the first 20 markers only
y2 <- haplosim(1000, which.snp=seq(20), x)

#
# Example 2
#

# simulate a sample of 500 haplotypes based on the ACE data set
set.seed(100)
data(ACEdata)
n <- 500
x <- haplodata(ACEdata)
y <- haplosim(n, x)

# compute the haplotype frequencies
# an haplotype starts at markers 17 and ends at marker 22
freq1 <- haplofreqs(ACEdata, 17, 22)
freq2 <- haplofreqs(y$data, 17, 22)

# extract the set of haplotypic configurations that are shared
# by real and simulated data and their frequencies
commonhapls <- intersect(names(freq1),names(freq2))
cfreq1 <- freq1[commonhapls]
cfreq2 <- freq2[commonhapls]

# compare real vs simulated haplotype frequencies
par(mar=c(10.1, 4.1, 4.1, 2.1), xpd=TRUE)
legend.text <- names(cfreq1)
bp <- barplot(cbind(cfreq1,cfreq2), main="Haplotype Frequencies",
       names.arg=c("Real","Simulated"), col=heat.colors(length(legend.text)))
legend(mean(range(bp)), -0.3, legend.text, xjust = 0.5,
       fill=heat.colors(length(legend.text)), horiz = TRUE)
chisq.test(x=n*cfreq2, p=cfreq1, simulate.p.value = TRUE, rescale.p = TRUE)
```

ldplot *LD plot*

Description

Creates a linkage disequilibrium plot from a matrix of pair-wise LD coefficients

Usage

```
ldplot(ld.mat, ld.type, color = heat.colors(50), title = NULL)
```

Arguments

ld.mat	A square matrix of LD coefficients
ld.type	A character value specifying what coefficients are used as input: either 'r' for correlation coefficients or 'd' for D/Dprime scores
color	A range of colors to be used for drawing. Default is heat.colors
title	Character string for the title of the plot

Author(s)

Giovanni Montana

References

Montana, G. HapSim: a simulation tool for generating haplotype data with pre-specified allele frequencies and LD coefficients. 2005.

Examples

```
data(ACEdata)

# LD plot of ACEdata using r^2 coefficients
ldplot(cor(ACEdata), ld.type='r')
```

`mergemats` *Merges two LD matrices*

Description

Merges two LD matrices. It can be used to compare the LD coefficients estimated in the real and simulated data sets

Usage

```
mergemats(mat1, mat2)
```

Arguments

`mat1`	First square matrix
`mat2`	Second square matrix of same dimensions

Value

The resulting matrix has upper triangular matrix from `mat1` and lower triangular matrix from `mat2`

Author(s)

Giovanni Montana

References

Montana, G. HapSim: a simulation tool for generating haplotype data with pre-specified allele frequencies and LD coefficients. 2005.

Index