Package: asymLD (via r-universe)

September 15, 2024

Type Package

Title Asymmetric Linkage Disequilibrium (ALD) for Polymorphic Genetic Data

Version 0.1

Date 2016-01-29

Author Richard M. Single

Maintainer Richard M. Single <rsingle@uvm.edu>

Suggests haplo.stats, fields

LazyLoad Yes

Description Computes asymmetric LD measures (ALD) for multi-allelic genetic data. These measures are identical to the correlation measure (r) for bi-allelic data.

License GPL-2

NeedsCompilation no

Date/Publication 2016-01-30 19:00:02

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

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

RemoteRef HEAD

RemoteSha d6d57c70268f048705645a4c8e9d2d091f3018f7

Contents

`asymLD-package` *asymLD: a package for computing asymmetric Linkage Disequilibrium (ALD) between two polymorphic genetic loci*

Description

Computes asymmetric LD measures (ALD) for polymorphic genetic data. These measures are identical to the correlation measure (r) for bi-allelic data.

Details

Package:	asymLD
Type:	Package
Version:	0.1
Date:	2015-03-13
License:	GPL-2

asymLD functions

The function compute.ALD() calculates asymmetric LD for haplotype frequency data. The function compute.AShomz() calculates allele specific homozygosity values for haplotype frequency data.

References

Thomson G, Single RM. Conditional Asymmetric Linkage Disequilibrium (ALD): Extending the Bi-Allelic r^2 Measure. Genetics. 2014 198(1):321-331. PMID:25023400

`compute.ALD` *Compute ALD.*

Description

A function to compute asymmetric Linkage Disequilibrium measures (ALD) for polymorphic genetic data. These measures are identical to the correlation measure (r) for bi-allelic data.

Usage

```
compute.ALD(dat, tolerance = 0.01)
```

Arguments

`dat` A data.frame with 5 required variables (having the names listed below):

`haplo.freq`	A numeric vector of haplotype frequencies.
`locus1`	A character vector indentifying the first locus.
`locus2`	A character vector indentifying the second locus.
`allele1`	A character vector indentifying the allele at locus 1.
`allele2`	A character vector indentifying the allele at locus 2.

`tolerance` A threshold for the sum of the haplotype frequencies. If the sum of the haplotype frequencies is greater than 1+tolerance or less than 1-tolerance an error is returned. The default is 0.01.

Value

The return value is a dataframe with the following components:

`locus1`	The name of the first locus.
`locus2`	The name of the second locus.
`F.1`	Homozygosity (expected under HWP) for locus 1.
`F.1.2`	Conditional homozygosity* for locus1 given locus2.
`F.2`	Homozygosity (expected under HWP) for locus 2.
`F.2.1`	Conditional homozygosity* for locus2 given locus1.
`ALD.1.2`	Asymmetric LD for locus1 given locus2.
`ALD.2.1`	Asymmetric LD for locus2 given locus1.

*Overall weighted haplotype-specific homozygosity for the first locus given the second locus.

Details

A warning message is given if the sum of the haplotype frequencies is greater than 1.01 or less than 0.99 (regardless of the `tolerance` setting). The haplotype frequencies that are passed to the function are normalized within the function to sum to 1.0 by dividing each frequency by the sum of the passed frequencies.

Examples

```
library(asymLD)

# An example using haplotype frequencies from Wilson(2010)
data(hla.freqs)
hla.a_b <- hla.freqs[hla.freqs$locus1=="A" & hla.freqs$locus2=="B",]
compute.ALD(hla.a_b)
hla.freqs$locus <- paste(hla.freqs$locus1, hla.freqs$locus2, sep="-")
compute.ALD(hla.freqs[hla.freqs$locus=="C-B",])
# Note: additonal columns on the input dataframe (e.g., "locus" above) are allowed, but
# ignored by the function.
```

```
# An example using genotype data from the haplo.stats package
require(haplo.stats)
data(hla.demo)
geno <- hla.demo[,5:8]  #DPB-DPA
label <- unique(gsub(".a(1|2)", "", colnames(geno)))
label <- paste("HLA*",label,sep="")
keep <- !apply(is.na(geno) | geno==0, 1, any)
em.keep  <- haplo.em(geno=geno[keep,], locus.label=label)
hapfreqs.df <- cbind(em.keep$haplotype, em.keep$hap.prob)
#format dataframe for ALD function
names(hapfreqs.df)[dim(hapfreqs.df)[2]] <- "haplo.freq"
names(hapfreqs.df)[1] <- "allele1"
names(hapfreqs.df)[2] <- "allele2"
hapfreqs.df$locus1 <- label[1]
hapfreqs.df$locus2 <- label[2]
head(hapfreqs.df)
compute.ALD(hapfreqs.df)
# Note that there is substantially less variablity (higher ALD) for HLA*DPA1
# conditional on HLA*DPB1 than for HLA*DPB1 conditional on HLA*DPA1, indicating
# that the overall variation for DPA1 is relatively low given specific DPB1 alleles

# An example using SNP data where results are symmetric and equal to the ordinary
# correlation measure (r)
data(snp.freqs)
snps <- c("rs1548306", "rs6923504", "rs4434496", "rs7766854")
compute.ALD(snp.freqs[snp.freqs$locus1==snps[2] & snp.freqs$locus2==snps[3],])

snp.freqs$locus <- paste(snp.freqs$locus1, snp.freqs$locus2, sep="-")
by(snp.freqs,list(snp.freqs$locus),compute.ALD)

# SNP1 & SNP2 : the r correlation & ALD measures are equivalent due to symmetry for
# bi-allelic SNPs
p.AB <- snp.freqs$haplo.freq[1]
p.Ab <- snp.freqs$haplo.freq[2]
p.aB <- snp.freqs$haplo.freq[3]
p.ab <- snp.freqs$haplo.freq[4]
p.A <- p.AB + p.Ab
p.B <- p.AB + p.aB
r.squared <- (p.AB - p.A*p.B)^2 / (p.A*(1-p.A)*p.B*(1-p.B))
sqrt(r.squared) #the r correlation measure
compute.ALD(snp.freqs[snp.freqs$locus1==snps[1] & snp.freqs$locus2==snps[2],])
```

`compute.AShomz` *Compute allele specific homozygosity.*

Description

A function to compute allele specific homozygosity values for haplotype frequency data. The allele specific homozygosity is the homozygosity statistic computed for alleles at one locus that are found on haplotypes with a specific allele (the focal allele) at the other locus (the focal locus).

Usage

```
compute.AShomz(dat, tolerance = 0.01, sort.var = c("focal", "allele"),
  sort.asc = rep(TRUE, length(sort.var)))
```

Arguments

dat	A data.frame with 5 required variables (having the names listed below):
haplo.freq	A numeric vector of haplotype frequencies.
locus1	A character vector indentifying the first locus.
locus2	A character vector indentifying the second locus.
allele1	A character vector indentifying the allele at locus 1.
allele2	A character vector indentifying the allele at locus 1.
tolerance	A threshold for the sum of the haplotype frequencies. If the sum of the haplotype frequencies is greater than 1+tolerance or less than 1-tolerance an error is returned. The default is 0.01.
sort.var	a vector of variable names specifying the "sort by" variables. The default is c("focal","allele").
sort.asc	a vector of TRUE/FALSE values, with the same length as "sort.var", indicating whether sorting of each variable is in ascending order. The default order is ascending.

Value

The return value is a dataframe with the following components:

loci	The locus names separated by "-".
focal	The name of the focal locus (locus conditioned on).
allele	The name of the focal allele (allele conditioned on).
allele.freq	The frequency of the focal allele.
as.homz	The allele specific homozygosity (on haplotypes with the focal allele).

Details

A warning message is given if the sum of the haplotype frequencies is greater than 1.01 or less than 0.99 (regardless of the tolerance setting). The haplotype frequencies that are passed to the function are normalized within the function to sum to 1.0 by dividing each frequency by the sum of the passed frequencies.

Examples

```
library(asymLD)

# An example using haplotype frequencies from Wilson(2010)
data(hla.freqs)
```

```
hla.dr_dq <- hla.freqs[hla.freqs$locus1=="DRB1" & hla.freqs$locus2=="DQB1",]
compute.ALD(hla.dr_dq)
compute.AShomz(hla.dr_dq, sort.var=c("focal","allele"), sort.asc=c(TRUE,TRUE))
compute.AShomz(hla.dr_dq, sort.var=c("focal","allele.freq"), sort.asc=c(FALSE,FALSE))
# Note that there is substantially less variablity (higher ALD) for HLA*DQB1
# conditional on HLA*DRB1 than for HLA*DRB1 conditional on HLA*DQB1, indicating
# that the overall variation for DQB1 is relatively low given specific DRB1 alleles.
# The largest contributors to ALD{DQB1|DRB1} are the DRB1*0301 and DRB1*1501 focal
# alleles, which have high allele frequencies and also have high allele specific
# homozygosity values.
```

`hla.freqs` *Pairwise haplotype frequencies for 7 HLA loci*

Description

HLA haplotype frequencies for 21 pairs of HLA loci in a set of 300 controls from a study of myopericarditis incidence following smallpox vaccination.

Usage

```
data(hla.freqs)
```

Format

A data frame with 3063 observations on the following 5 variables.

`haplo.freq` a numeric vector

`locus1` a character vector

`locus2` a character vector

`allele1` a character vector

`allele2` a character vector

Source

Wilson, C., 2010 Identifying polymorphisms associated with risk for the development of myopericarditis following smallpox vaccine. The Immunology Database and Analysis Portal (ImmPort), Study #26.

References

https://immport.niaid.nih.gov/immportWeb/clinical/study/displayStudyDetails.do?itemList=SDY26

`lsort` *Sort a data.frame.*

Description

A function to sort a data.frame on specific columns.

Usage

```
lsort(dat, by = 1:dim(dat)[2], asc = rep(TRUE, length(by)),
  na.last = TRUE)
```

Arguments

`dat`	a dataframe or a matrix ("dimnames" are used as the variable names for a matrix)
`by`	a vector or a list of variable names or column indices specifying the "sort by" variables, the default is to sort by all variables in the order they appear in the data set.
`asc`	a vector with the same length as "by" indicating whether the sorting of each "by" variable is in ascending order, the default order is ascending.
`na.last`	a flag indicating whether missing values are placed as the last elements in the data set, the default is TRUE

Value

The return value is a sorted dataframe.

Details

The input dataframe is not modified. The code is adapted from code posted to an old s-news listserve.

Examples

```
## Not run:
library(asymLD)
data(snp.freqs)

# sort snp.freqs by "locus1" (ascending) and "allele1" (descending)
newdata <- lsort(snp.freqs, by=c("locus1","allele1"), asc=c(T,F))
head(newdata)
# sort snp.freqs by the fourth and the second variable (ascending)
newdata <- lsort(snp.freqs, by=c(4,2))
# sort "snp.freqs" by "locus1" and the 5th variable (ascending)
newdata <- lsort(snp.freqs, by=list("locus1",5))


## End(Not run)
```

snp.freqs *Pairwise haplotype frequencies for 4 SNP loci*

Description

HLA haplotype frequencies for 6 pairs of SNP loci from the de Bakker et al. 2006 data for 90 unrelated individuals with European ancestry (CEU) from the Centre d'Etude du Polymorphisme Humain (CEPH) collection obtained from the Tagger/MHC webpage.

Usage

```
data(snp.freqs)
```

Format

A data frame with 20 observations on the following 5 variables.

`locus1` a character vector

`locus2` a character vector

`allele1` a character vector

`allele2` a character vector

`haplo.freq` a numeric vector

Details

For bi-allelic SNP data the ALD measures are symmetric and equivalent to the r correlation measure of LD.

Source

de Bakker, P. I., G. McVean, P. C. Sabeti, M. M. Miretti, T. Green et al., 2006 A high-resolution HLA and SNP haplotype map for disease association studies in the extended human MHC. Nat.Genet. 38: 1166-1172.

References

http://www.broadinstitute.org/mpg/tagger/mhc.html

Index