Package 'hscovar'

March 4, 2020

Type Package

Title Calculation of Covariance Between Markers for Half-Sib Families

Version 0.1.2

Description The theoretical covariance between pairs of markers is calculated from either paternal haplotypes and maternal linkage disequilibrium (LD) or vise versa. A genetic map is required. Grouping of markers is based on the correlation matrix and a representative marker is suggested for each group. The implementation relies on paternal half-sib families and biallelic markers. If maternal half-sib families are used, the roles of sire/dam are swapped. Multiple families can be considered.
Wittenburg, Bonk, Doschoris, Reyer (2019) ``Design of Experiments for Fine-Mapping Quantitative Trait Loci in Livestock Populations" <doi:10.1101/2019.12.17.879106>.
Carlson, Eberle, Rieder, Yi, Kruglyak, Nickerson (2004) ``Selecting a maximally informative set of single-nucleotide polymorphisms for association analyses using linkage disequilibrium" <doi:10.1086/381000>.

Depends R (>= 3.5.0)

Imports parallel, Matrix, foreach, rlist

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

NeedsCompilation no

Author Dörte Wittenburg [aut, cre],
Michael Doschoris [aut],
Jan Klosa [ctb]

Maintainer Dörte Wittenburg <wittenburg@fbn-dummerstorf.de>

Repository CRAN

Date/Publication 2020-03-04 10:00:02 UTC

R topics documented:

`CovarMatrix` *CovarMatrix*

Description

Calculation of covariance matrices from LD-matrices

Usage

```
CovarMatrix(exp_freq_mat, LDDam, LDSire, Ns)
```

Arguments

`exp_freq_mat`	[MATRIX] paternal EXPECTATION matrix
`LDDam`	[MATRIX] maternal Linkage Disequilibrium matrix
`LDSire`	[LIST] Linkage disequilibrium matrices for the sires; each element of the list corresponds to a family
`Ns`	[VECTOR] family size for each sire s

Details

The internal suMM function works on lists!

Value

`covK` (p x p) matrix of covariance between markers

Examples

```
data(testdata)
G <- Haplo2Geno(H.sire)
E <- ExpectMat(G)
LDfam2 <- LDsire(H.sire, pos.chr, family = 3:4)
LDfam3 <- LDsire(H.sire, pos.chr, family = 5:6)
## covariance matrix based on sires 2 and 3 only, each with 100 progeny
K <- CovarMatrix(E[2:3, ], LDDam = matLD, LDSire = list(LDfam2, LDfam3), Ns = c(100, 100))
```

`CovMat` *CovMat*

Description

Calculation of covariance or correlation matrix

Usage

```
CovMat(linkDam, haploSire, nfam, pos_chr, corr = T)
```

Arguments

`linkDam`	(p x p) matrix of maternal LD between pairs of p markers; matrix is block diagonal in case of multiple chromosomes
`haploSire`	(2N x p) matrix of sires haplotypes for all chromosomes (2 lines per sire)
`nfam`	vector (LEN N) containing number of progeny per sire or scalar value in case of equal family size
`pos_chr`	list (LEN number of chromosomes) of vectors (LEN number of markers) of genetic positions in Morgan per chromosome
`corr`	logical; `TRUE` (default) if output is correlation matrix or `FALSE` if output is covariance matrix

Details

The theoretical covariance between pairs of markers is calculated from either paternal haplotypes and maternal linkage disequilibrium (LD) or vise versa. A genetic map is required. The implementation relies on paternal half-sib families and biallelic markers such as single nucleotide polymorphisms (SNP).

If maternal half-sib families are used, the roles of sire/dam are swapped. Multiple families can be considered.

Value

list (LEN 2) of matrix (DIM $p1$ x $p1$) and vector (LEN $p1$) with $p1 \leq p$

`K` covariance matrix OR

`R` correlation matrix

`valid.snps` vector of SNP indices considered for covariance/ correlation matrix

Note

If you have maternal haplotypes (H.mothers; same format as H.sire) instead of maternal LD (matLD) then LD can be estimated from counting haplotype frequencies as:

`matLD <-LDdam(inMat = H.mother,pos.chr)`

If multiple chromosomes are considered, then, for instance:

`pos.chr <-list(pos.snp.chr1,pos.snp.chr2,pos.snp.chr3)`

References

Wittenburg, Bonk, Doschoris, Reyer (2019) "Design of Experiments for Fine-Mapping Quantitative Trait Loci in Livestock Populations" https://doi.org/10.1101/2019.12.17.879106

Examples

```
### 1: INPUT DATA
data(testdata)
### 2: COVARIANCE/CORRELATION MATRIX
corrmat <- CovMat(matLD, H.sire, 100, pos.chr, corr = TRUE)
### 3: TAGSNPS FROM CORRELATION MATRIX
bin <- tagSNP(corrmat$R)
bin <- tagSNP(corrmat$R, 0.5)
```

`ExpectMat` *ExpectMat*

Description

Expected value of paternally inherited allele

Usage

```
ExpectMat(inMat)
```

Arguments

`inMat`	[MATRIX] The paternal genotype matrix

Value

`ExP.Fa` (N x p) matrix of expected values

Examples

```
data(testdata)
G <- Haplo2Geno(H.sire)
E <- ExpectMat(G)
```

H.sire *testdata: sire haplotypes*

Description

(2N x p) matrix of sire haplotypes for all chromosomes (2 lines per sire); unknown alleles are marked as 9

Usage

```
H.sire
```

Format

An object of class `matrix` with 10 rows and 300 columns.

Haplo2Geno *Haplo2Geno*

Description

Conversion of haplotypes into genotypes (without checking for missing values)

Usage

```
Haplo2Geno(inpMat)
```

Arguments

`inpMat`	[MATRIX] haplotype matrix (2 lines per individual)

Value

`outMa` (N x p) genotype matrix

Examples

```
data(testdata)
G <- Haplo2Geno(H.sire)
```

LDdam *LDdam*

Description

CALCULATE THE LINKAGE DISEQUILIBRIUM MATRIX FOR THE DAMS

Usage

```
LDdam(inMat, pos_chr)
```

Arguments

inMat	[MATRIX] The maternal HAPLOTYPE matrix.
pos_chr	[LIST] The marker positions in Morgan on chromosomes.

Details

The function generates a block diagonal sparse matrix based on Matrix::bdiag. Use as.matrix() to obtain a regular one.

Value

Dd (p x p) matrix of maternal LD

Examples

```
## haplotype matrix of n individuals at p SNPs
p <- 10; n <- 4
mat <- matrix(ncol = p, nrow = 2 * n, sample(c(0, 1), size = 2 * n * p, replace = TRUE))
LDdam(mat, list(1:p))
```

LDsire *LDsire*

Description

CALCULATE THE LINKAGE DISEQUILIBRIUM MATRIX FOR THE SIRES

Usage

```
LDsire(inMat, pos_chr, family, map_fun = "haldane")
```

Arguments

inMat	[MATRIX] Haplotype matrix for sires for all chromosomes.
pos_chr	[LIST] The marker positions in Morgan on chromosomes.
family	[VECTOR] Which family (sire) should be processed? Vector with consecutive entries of the form 1:2, 3:4, 5:6 and so on, linking to haplotypes (rows in inMat) of the corresponding sire
map_fun	[haldane, kosambi] The mapping function applied.

Details

The function generates a block diagonal sparse matrix based on Matrix::bdiag. Use as.matrix() to obtain a regular one.

Value

Ds

Ds (p x p) matrix of paternal LD

Examples

```
data(testdata)
LDfam2 <- LDsire(H.sire, pos.chr, family = 3:4)
```

---

matLD *testdata: maternal linkage disequilibrium*

Description

(p x p) matrix of maternal LD between pairs of p markers; matrix is block diagonal in case of multiple chromosomes

Usage

```
matLD
```

Format

An object of class matrix with 300 rows and 300 columns.

pos.chr *testdata: genetic map positions*

Description

list of vectors of genetic map positions per chromosome

Usage

```
pos.chr
```

Format

An object of class `list` of length 1.

tagSNP *tagSNP*

Description

Grouping of markers depending on correlation structure

Usage

```
tagSNP(mat, threshold = 0.8)
```

Arguments

`mat`	(p x p) correlation matrix
`threshold`	lower value of correlation considered for grouping

Details

Grouping of markers is based on the correlation matrix. Apart from this, the strategy for grouping is similar to Carlson et al. (2004). A representative marker is suggested for each group.

Value

list (LEN number of groups) of lists (LEN 2); marker names correspond to column names of mat

`snps` vector of marker IDs in group

`tagsnp` representative marker suggested for this group

References

Carlson, C. S., Eberle, M. A., Rieder, M. J., Yi, Q., Kruglyak, L. & Nickerson, D. A. Selecting a maximally informative set of single- nucleotide polymorphisms for association analyses using linkage disequilibrium. Am. J. Hum. Genet., 2004, 74:106-120.

Examples

```
### 1: INPUT DATA
data(testdata)
### 2: COVARIANCE/CORRELATION MATRIX
corrmat <- CovMat(matLD, H.sire, 100, pos.chr, corr = TRUE)
### 3: TAGSNPS FROM CORRELATION MATRIX
bin <- tagSNP(corrmat$R)
bin <- tagSNP(corrmat$R, 0.5)
as.numeric(unlist(rlist::list.select(bin, tagsnp)))
```

---

testdata *Description of the testdata*

Description

The data set contains paternal haplotypes, maternal LD and genetic map positions that are required to calculate the covariance between pairs of markers.

The raw data can be downloaded at the source given below. Then, executing the following R code leads to the data that have been provided as `testdata.RData`.

H.sire (2N x p) haplotype matrix for sires for all chromosomes (2 lines per sire)

matLD (p x p) matrix of maternal LD between pairs of p markers; matrix is block diagonal in case of multiple chromosomes

pos.chr list of vectors of genetic map positions per chromosome

Source

The data are available from the RADAR repository https://dx.doi.org/10.22000/280

Examples

```
## data.frame of estimates of paternal recombination rate and maternal LD
load('Result.RData')
## list of haplotypes of sires for each chromosome
load('sire_haplotypes.RData')
## physical map
map <- read.table('map50K_ARS_reordered.txt', header = T)
## select target region
chr <- 1
window <- 301:600
## map information of target region
```

```
map.target <- map[map$Chr == chr, ][window, ]
Result.target <- Result[(Result$Chr == chr) & (Result$SNP1 %in% window) &
  (Result$SNP2 %in% window), ]
## SNP position in Morgan approximated from recombination rate
part <- Result.target[Result.target$SNP1 == window[1], ]
sp <- smooth.spline(x = map.target$locus_Mb[part$SNP2 - window[1] + 1], y = part$Theta, df = 4)
pos.snp <- predict(sp, x =  map.target$locus_Mb[window - window[1] + 1])$y
## list of SNPs positions
pos.chr <- list(pos.snp)
## haplotypes of sires (mating candidates) in target region
H.sire <- rlist::list.rbind(haps[[chr]])[, window]
## matrix of maternal LD (block diagonal if multiple chromosome)
matLD <- matrix(0, ncol = length(window), nrow = length(window))
## off-diagonal elements
for(l in 1:nrow(Result.target)){
  id1 <- Result.target$SNP1[l] - window[1] + 1
  id2 <- Result.target$SNP2[l] - window[1] + 1
  matLD[id1, id2] <- matLD[id2, id1] <- Result.target$D[l]
}
## diagonal elements
for(k in unique(Result.target$SNP1)){
  id <- k - window[1] + 1
 p <- Result.target$fAA[Result.target$SNP1 == k] + Result.target$fAB[Result.target$SNP1 == k]
  matLD[id, id] <- max(p * (1 - p))
}
```

Index