Package ‘hsrecombi’

March 23, 2021

Type Package
Title Estimation of Recombination Rate and Maternal LD in Half-Sibs
Version 0.3.4
Date 2021-03-23
Description Paternal recombination rate and maternal linkage disequilibrium (LD) are estimated for pairs of biallelic markers such as single nucleotide polymorphisms (SNPs) from progeny genotypes and sire haplotypes. The implementation relies on paternal half-sib families. If maternal half-sib families are used, the roles of sire/dam are swapped. Multiple families can be considered. For parameter estimation, at least one sire has to be double heterozygous at the investigated pairs of SNPs. Based on recombination rates, genetic distances between markers can be estimated. Markers with unusually large recombination rate to markers in close proximity can be discarded in this derivation.
*A pipeline is available at github*
<https://github.com/wittenburg/hsrecombi>

Depends R (>= 3.5.0)
Imports Rcpp (>= 1.0.3), hsphase, dplyr, data.table, rlist, quadprog, curl
License GPL (>= 2)
Encoding UTF-8
LazyData true
LinkingTo Rcpp
RoxygenNote 7.1.1
Suggests knitr, rmarkdown, formatR, AlphaSimR (>= 0.13.0), doParallel, ggplot2
VignetteBuilder knitr
checkCandidates

Candidates for misplacement

Description
Search for SNPs with unusually large estimates of recombination rate

Usage
checkCandidates(final, win = 30, quant = 0.99)

Arguments
- **final**: table of results produced by `editraw` with pairwise estimates of recombination rate between p SNPs within chromosome; minimum required data frame with columns SNP1, SNP2 and theta
- **win**: optional value for window size; default value 30
- **quant**: optional value; default value 0.99, see details
Details

Markers with unusually large estimates of recombination rate to close SNPs are candidates for misplacements in the underlying assembly. The mean of recombination rate estimates with win subsequent or preceding markers is calculated and those SNPs with mean value exceeding the quart quantile are denoted as candidates which have to be manually curated! This can be done, for instance, by visual inspection of a correlation plot containing estimates of recombination rate in a selected region.

Value

vector of SNP indices for further verification

References


Examples

```r
### test data
data(targetregion)
### make list for paternal half-sib families
hap <- makehaplist(daughterSire, hapSire)
### parameter estimates on a chromosome
res <- hsrecombi(hap, genotype.chr, map.chr$SNP)
### post-processing to achieve final and valid set of estimates
final <- editraw(res, map.chr)
### check for candidates of misplacement
snp <- checkCandidates(final)
```

countNumbers

| countNumbers | Count genotype combinations at 2 SNPs |

Description

Count genotype combinations at 2 SNPs

Arguments

- `X` numeric matrix of genotypes

Value

count vector of counts of 9 possible genotypes at SNP pair
daughterSire: allocation of paternal half-sib families

**Description**
Vector of sire ID for each progeny

**Usage**
daughterSire

**Format**
An object of class `integer` of length 265.

editraw: Editing results of hsrecombi

**Description**
Process raw results from hsrecombi, decide which out of two sets of estimates is more likely and prepare list of final results

**Usage**
editraw(Roh, map1)

**Arguments**
- `Roh`: list of raw results from hsrecombi
- `map1`: data.frame containing information on physical map, at least:
  - SNP: SNP ID
  - locus_Mb: physical position in Mbp of SNP on chromosomes
  - Chr: chromosome of SNP

**Value**
final table of results

- SNP1: index 1. SNP
- SNP2: index 2. SNP
- D: maternal LD
- fAA: frequency of maternal haplotype 1-1
- fAB: frequency of maternal haplotype 1-0
### geneticPosition

- **fBA**: frequency of maternal haplotype 0-1
- **fBB**: frequency of maternal haplotype 0-0
- **p1**: Maternal allele frequency (allele 1) SNP1
- **p2**: Maternal allele frequency (allele 1) SNP2
- **nfam1**: size of genomic family 1
- **nfam2**: size of genomic family 2
- **error**: 0 if computations were without error; 1 if EM algorithm did not converge
- **iteration**: number of EM iterations
- **theta**: paternal recombination rate
- **r2**: $r^2$ of maternal LD
- **logL**: value of log likelihood function
- **unimodal**: 1 if likelihood is unimodal; 0 if likelihood is bimodal
- **critical**: 0 if parameter estimates were unique; 1 if parameter estimates were obtained via decision process
- **locus_Mb**: physical distance between SNPs in Mbp

### Examples

```r
### test data
data(targetregion)
### make list for paternal half-sib families
hap <- makehaplist(daughterSire, hapSire)
### parameter estimates on a chromosome
res <- hsrecombi(hap, genotype.chr, map.chr$SNP)
### post-processing to achieve final and valid set of estimates
final <- editraw(res, map.chr)
```

#### geneticPosition

<table>
<thead>
<tr>
<th>geneticPosition</th>
<th>Estimation of genetic position</th>
</tr>
</thead>
</table>

### Description

Estimation of genetic positions (in centi Morgan)

### Usage

```r
geneticPosition(final, exclude = NULL, threshold = 0.05)
```

### Arguments

- **final**: table of results produced by editraw with pairwise estimates of recombination rate between p SNPs within chromosome; minimum required data frame with columns SNP1, SNP2 and theta
- **exclude**: optional vector (LEN q) of SNPs to be excluded (e.g., candidates of misplaced SNPs)
- **threshold**: optional value; recombination rates $\leq$ threshold are considered for smoothing
Details

Smoothing of recombination rates (theta) \( \leq 0.05 \) via quadratic optimization provides an approximation of genetic distances (in Morgan) between SNPs. The cumulative sum \( \times 100 \) yields the genetic positions in cM.

The minimization problem \((\text{theta} - D\ d)^2\) is solved s.t. \( d > 0 \) where \( d \) is the vector of genetic distances between adjacent markers but theta is not restricted to adjacent markers. The incidence matrix \( D \) contains 1’s for those intervals contributing to the total distance relevant for each theta.

Estimates of theta = 1e-6 are neglected as these values coincide with start values and indicate that (because of a very flat likelihood surface) no meaningful estimate of recombination rate has been obtained.

Value

vector (LEN p) of genetic positions of SNPs (in cM)

References


Examples

```r
### test data
data(targetregion)
### make list for paternal half-sib families
hap <- makehaplist(daughterSire, hapSire)
### parameter estimates on a chromosome
res <- hsrecombi(hap, genotype.chr, map.chr$SNP)
### post-processing to achieve final and valid set of estimates
final <- editraw(res, map.chr)
### approximation of genetic positions
pos <- geneticPosition(final)
```

Description

matrix of progeny genotypes in target region on chromosome BTA1

Usage

genotype.chr

Format

An object of class matrix (inherits from array) with 265 rows and 300 columns.
**hapSire**

**Description**

matrix of sire haplotypes in target region on chromosome BTA1

**Usage**

hapSire

**Format**

An object of class matrix (inherits from array) with 10 rows and 301 columns.

---

**hsrecombi**

**Estimation of recombination rate and maternal LD**

**Description**

Wrapper function for estimating recombination rate and maternal linkage disequilibrium between intra-chromosomal SNP pairs by calling EM algorithm

**Usage**

hsrecombi(hap, genotype.chr, snp.chr, only.adj = FALSE, prec = 1e-06)

**Arguments**

- **hap**: list (LEN 2) of lists
  - famID: list (LEN number of sires) of vectors (LEN n.progeny) of progeny indices relating to lines in genotype matrix
  - sireHap: list (LEN number of sires) of matrices (DIM 2 x p) of sire haplotypes (0, 1) on investigated chromosome
- **genotype.chr**: matrix (DIM n x p) of all progeny genotypes (0, 1, 2) on a chromosome with p SNPs
- **snp.chr**: vector(LEN p) of SNP indices as in physical map
- **only.adj**: logical; if TRUE, recombination rate is calculated only between neighbouring markers
- **prec**: scalar; precision of estimation
Details

Paternal recombination rate and maternal linkage disequilibrium (LD) are estimated for pairs of biallelic markers (such as single nucleotide polymorphisms; SNPs) from progeny genotypes and sire haplotypes. At least one sire has to be double heterozygous at the investigated pairs of SNPs. All progeny are merged in two genomic families: (1) coupling phase family if sires are double heterozygous 0-0/1-1 and (2) repulsion phase family if sires are double heterozygous 0-1/1-0. So far it is recommended processing the chromosomes separately. If maternal half-sib families are used, the roles of sire/dam are swapped. Multiple families can be considered.

Value

list (LEN p - 1) of data.frames; for each SNP, parameters are estimated with all following SNPs; two solutions (prefix sln1 and sln2) are obtained for two runs of the EM algorithm

SNP1 index 1. SNP
SNP2 index 2. SNP
D maternal LD
fAA frequency of maternal haplotype 1-1
fAB frequency of maternal haplotype 1-0
fBA frequency of maternal haplotype 0-1
fBB frequency of maternal haplotype 0-0
p1 Maternal allele frequency (allele 1)
p2 Maternal allele frequency (allele 0)
nfam1 size of genomic family 1
nfam2 size of genomic family 2
error 0 if computations were without error; 1 if EM algorithm did not converge
iteration number of EM iterations
theta paternal recombination rate
r2 r^2 of maternal LD
logL value of log likelihood function
unimodal 1 if likelihood is unimodal; 0 if likelihood is bimodal
critical 0 if parameter estimates are unique; 1 if parameter estimates at both solutions are valid,
then decision process follows in post-processing function "editraw"

Afterwards, solutions are compared and processed with function editraw, yielding the final estimates for each valid pair of SNPs.

References


### Examples

```r
### test data
data(targetregion)
### make list for paternal half-sib families
hap <- makehaplist(daughterSire, hapSire)
### parameter estimates on a chromosome
res <- hsrecombi(hap, genotype.chr, map.chr$SNP)
### post-processing to achieve final and valid set of estimates
final <- editraw(res, map.chr)
```

### Description

**Expectation Maximisation (EM) algorithm**

### Usage

```r
LDHScpp(XGF1, XGF2, fAA, fAB, fBA, theta, display, threshold)
```

### Arguments

- **XGF1**: numeric matrix of progeny genotypes in genomic family 1
- **XGF2**: numeric matrix of progeny genotypes in genomic family 2
- **fAA**: frequency of maternal haplotype 1-1
- **fAB**: frequency of maternal haplotype 1-0
- **fBA**: frequency of maternal haplotype 0-1
- **theta**: paternal recombination rate
- **display**: logical for displaying additional information
- **threshold**: convergence criterion

### Value

- list of parameter estimates

  - D: maternal LD
  - fAA: frequency of maternal haplotype 1-1
  - fAB: frequency of maternal haplotype 1-0
  - fBA: frequency of maternal haplotype 0-1
  - fBB: frequency of maternal haplotype 0-0
  - p1: Maternal allele frequency (allele 1)
  - p2: Maternal allele frequency (allele 0)
  - nfam1: size of genomic family 1
nfam2 size of genomic family 2
error 0 if computations were without error; 1 if EM algorithm did not converge
iteration number of EM iterations
theta paternal recombination rate
r2 r^2 of maternal LD
logL value of log likelihood function

loglikfun Calculate log-likelihood function

Description
Calculate log-likelihood function

Arguments
counts integer vector of observed 2-locus genotype
fAA frequency of maternal haplotype 1-1
fAB frequency of maternal haplotype 1-0
fBA frequency of maternal haplotype 0-1
fBB frequency of maternal haplotype 0-0
theta paternal recombination rate

Value
lik value of log likelihood at parameter estimates

makehap Make list of imputed sire haplotypes

Description
List of sire haplotypes is set up in the format required for hsrecombi. Sire haplotypes are imputed from progeny genotypes using R package hsphase.

Usage
makehap(sireID, daughterSire, genotype.chr, nmin = 30)
**makehaplist**

**Arguments**

- sireID: vector (LEN N) of IDs of all sires
- daughterSire: vector (LEN n) of sire ID for each progeny
- genotype.chr: matrix (DIM n x p) of progeny genotypes on a single chromosome with p SNPs
- nmin: scalar, minimum required number of progeny for proper imputation, default 30

**Value**

list (LEN 2) of lists. For each sire:

- famID: list (LEN N) of vectors (LEN n.progeny) of progeny indices relating to lines in genotype matrix
- sireHap: list (LEN N) of matrices (DIM 2 x p) of sire haplotypes (0, 1) on investigated chromosome

**References**


**Examples**

```r
data(targetregion)
hap <- makehap(unique(daughterSire), daughterSire, genotype.chr)
```

---

**makehaplist**

*Make list of sire haplotypes*

**Description**

List of sire haplotypes is set up in the format required for hsrecombi. Haplotypes (obtained by external software) are provided.

**Usage**

```r
makehaplist(daughterSire, hapSire, nmin = 1)
```

**Arguments**

- daughterSire: vector (LEN n) of sire ID for each progeny
- hapSire: matrix (DIM 2N x p + 1) of sire haplotype at p SNPs; 2 lines per sire, 1. columns contains sire ID
- nmin: scalar, minimum number of progeny required, default 1
Description

List of sire haplotypes is set up in the format required for hsrecombi. Sire haplotypes are imputed from progeny genotypes using R package hsphase. Furthermore, recombination rate estimates between adjacent SNPs from hsphase are reported.

Usage

makehappm(sireID, daughterSire, genotype.chr, nmin = 30)

Arguments

sireID vector (LEN N) of IDs of all sires
daughterSire vector (LEN n) of sire ID for each progeny
genotype.chr matrix (DIM n x p) of progeny genotypes on a single chromosome with p SNPs
nmin scalar, minimum number of progeny required, default 1

Value

hap list (LEN 2) of lists. For each sire:

famID list (LEN N) of vectors (LEN n.progeny) of progeny indices relating to lines in genotype matrix
sireHap list (LEN N) of matrices (DIM 2 x p) of sire haplotypes (0, 1) on investigated chromosome
probRec vector (LEN p - 1) of proportion of recombinant progeny over all families between adjacent SNPs
numberRec list (LEN N) of vectors (LEN n.progeny) of number of recombination events per animal
References


Examples

data(targetregion)
hap <- makehappm(unique(daughterSire), daughterSire, genotype.chr)

Description

SNP marker map in target region on chromosome BTA1 according to ARS-UCD1.2

Usage

map.chr

Arguments

map.chr  data frame
SNP     SNP index
Chr     chromosome of SNP
locus_bp  physical position of SNP in bp
locus_Mb  physical position of SNP in Mbp
markernamex official SNP name

Format

An object of class data.frame with 300 rows and 5 columns.
**startvalue**

*Start value for maternal allele and haplotype frequencies*

**Description**

Determine default start values for Expectation Maximisation (EM) algorithm that is used to estimate paternal recombination rate and maternal haplotype frequencies.

**Usage**

```
startvalue(Fam1, Fam2, Dd = 0, prec = 1e-06)
```

**Arguments**

- **Fam1**: matrix (DIM n.progeny x 2) of progeny genotypes of genomic family with coupling phase sires (1) at SNP pair.
- **Fam2**: matrix (DIM n.progeny x 2) of progeny genotypes of genomic family with repulsion phase sires (2) at SNP pair.
- **Dd**: maternal LD, default 0.
- **prec**: minimum accepted start value for fAA, fAB, fBA; default 1e-6.

**Value**

- list (LEN 8)
  - fAA.start: frequency of maternal haplotype 1-1.
  - fAB.start: frequency of maternal haplotype 1-0.
  - fBA.start: frequency of maternal haplotype 0-1.
  - p1: estimate of maternal allele frequency (allele 1) when sire is heterozygous at SNP1.
  - p2: estimate of maternal allele frequency (allele 1) when sire is heterozygous at SNP2.
  - L1: lower bound of maternal LD.
  - L2: upper bound for maternal LD.
  - critical: 0 if parameter estimates are unique; 1 if parameter estimates at both solutions are valid.

**Examples**

```r
n1 <- 100
n2 <- 20
G1 <- matrix(ncol = 2, nrow = n1, sample(c(0:2), replace = TRUE, size = 2 * n1))
G2 <- matrix(ncol = 2, nrow = n2, sample(c(0:2), replace = TRUE, size = 2 * n2))
startvalue(G1, G2)
```
Description

The data set contains sire haplotypes, assignment of progeny to sire, progeny genotypes and physical map information in a target region.

The raw data can be downloaded at the source given below. Then, executing the following R code leads to the data provided in targetregion.RData.

- **hapSire**: matrix of sire haplotypes of each sire; 2 lines per sire; 1. column contains sireID
- **daughterSire**: vector of sire ID for each progeny
- **genotype.chr**: matrix of progeny genotypes
- **map.chr**: SNP marker map in target region

Source

The data are available at RADAR doi: 10.22000/280

Examples

```r
## Not run:
# download data from RADAR (requires about 1.4 GB)
url <- "https://www.radar-service.eu/radar-backend/archives/fqSPQoIvjtOGJlav/versions/1/content"
curl_download(url = url, "tmp.tar")
untar("tmp.tar")
file.remove("tmp.tar")
path <- "10.22000-280/data/dataset"
## list of haplotypes of sires for each chromosome
load(file.path(path, "sire_haplotypes.RData"))
## assign progeny to sire
daughterSire <- read.table(file.path(path, "assign_to_family.txt"))[, 1]
## progeny genotypes
X <- as.matrix(read.table(file.path(path, "XFam-ARS.txt")))
## physical and approximated genetic map
map <- read.table(file.path(path, "map50K_ARS_reordered.txt"), header = T)
## select target region
chr <- 1
window <- 301:600
## map information of target region
map.chr <- map[map$Chr == chr, ][window, 1:5]
## matrix of sire haplotypes in target region
hapSire <- rlist::list.rbind(haps[[chr]])
sireID <- 1:length(unique(daughterSire))
hapSire <- cbind(rep(sireID, each = 2), hapSire[, window])
## matrix of progeny genotypes
genotype.chr <- X[, map.chr$SNP]

## End(Not run)
```
Index

* datasets
  * daughterSire, 4
  * genotype.chr, 6
  * hapSire, 7
  * map.chr, 13

checkCandidates, 2
countNumbers, 3
daughterSire, 4
editraw, 4
geneticPosition, 5
genotype.chr, 6
hapSire, 7
hsrecombi, 7
LDHScpp, 9
loglikfun, 10
makehap, 10
makehaplist, 11
makehapppm, 12
map.chr, 13
startvalue, 14
targetregion, 15