Package ‘hsrecombi’

June 7, 2023

Type Package

Title Estimation of Recombination Rate and Maternal LD in Half-Sibs

Version 1.0.1

Date 2023-06-07

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 (i.e. putatively misplaced markers) shall be discarded in this derivation.
A workflow description is attached as vignette.
A pipeline is available at GitHub
<`https://github.com/wittenburg/hsrecombi`>
Hampel, Teuscher, Gomez-Raya, Doschoris, Wittenburg (2018) ``Estimation of recombination rate and maternal linkage disequilibrium in half-sibs'' <doi:10.3389/fgene.2018.00186>.
Gomez-Raya (2012) ``Maximum likelihood estimation of linkage disequilibrium in half-sib families'' <doi:10.1534/genetics.111.137521>.

Depends R (>= 3.5.0)

Imports Rcpp (>= 1.0.3), hsphase, dplyr, data.table, rlist, quadprog, curl, Matrix

License GPL (>= 2)

Encoding UTF-8

LazyData true

LinkingTo Rcpp

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, formatR, AlphaSimR (>= 0.13.0), doParallel, ggplot2

VignetteBuilder knitr

Language en-GB

NeedsCompilation yes

Author Dörte Wittenburg [aut, cre]

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

Repository CRAN

Date/Publication 2023-06-07 08:20:06 UTC

R topics documented:

bestmapfun 2
checkCandidates 3
countNumbers 5
daughterSire 5
editraw 5
felsenstein 7
geneticPosition 7
genotype.chr 9
haldane 9
hapSire 10
hsrecombi 10
karlin 12
kosambi 13
LDHScpp 13
loglikfun 14
makehap 15
makehaplist 16
makehappm 17
map.chr 18
rao 18
rao inverse 19
startvalue 20
targetregion 21

Index **23**

bestmapfun *Best fitting genetic-map function*

Description

Approximation of mixing parameter of system of map functions

Usage

```
bestmapfun(theta, dist_M)
```

Arguments

`theta`	vector of recombination rates
`dist_M`	vector of genetic positions

Details

The genetic mapping function that fits best to the genetic data (recombination rate and genetic distances) is obtained from Rao's system of genetic-map functions. The corresponding mixing parameter is estimated via 1-dimensional constrained optimisation. See vignette for its application to estimated data.

Value

list (LEN 2)

mixing mixing parameter of system of genetic mapping functions

mse minimum value of target function (theta - dist_M)^2

References

Rao, D.C., Morton, N.E., Lindsten, J., Hulten, M. & Yee, S (1977) A mapping function for man. Human Heredity 27: 99-104. doi: 10.1159/000152856

Examples

```
theta <- seq(0, 0.5, 0.01)
gendist <- -log(1 - 2 * theta) / 2
bestmapfun(theta, gendist)
```

`checkCandidates` *Candidates for misplacement*

Description

Search for SNPs with unusually large estimates of recombination rate

Usage

```
checkCandidates(final, map1, 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`

`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

`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 misplacement 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 `quant` 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 IDs for further verification

References

Hampel, A., Teuscher, F., Gomez-Raya, L., Doschoris, M. & Wittenburg, D. (2018) Estimation of recombination rate and maternal linkage disequilibrium in half-sibs. Frontiers in Genetics 9:186. doi: 10.3389/fgene.2018.00186

Examples

```
### 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)
### post-processing to achieve final and valid set of estimates
final <- editraw(res, map.chr)
### check for candidates of misplacement
snp <- checkCandidates(final, map.chr)
```

`countNumbers` *Count genotype combinations at 2 SNPs*

Description

Count genotype combinations at 2 SNPs

Arguments

`X` integer matrix of genotypes

Value

`count` vector of counts of 9 possible genotypes at SNP pair

`daughterSire` *targetregion: 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

fBA frequency of maternal haplotype 0-1

fBB frequency of maternal haplotype 0-0

p1 Maternal allele frequency (allele 1) at SNP1

p2 Maternal allele frequency (allele 1) at 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

```
### 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)
### post-processing to achieve final and valid set of estimates
final <- editraw(res, map.chr)
```

felsenstein *Felsenstein's genetic map function*

Description

Calculation of genetic distances from recombination rates given an interference parameter

Usage

```
felsenstein(K, x, inverse = F)
```

Arguments

`K`	parameter (numeric) corresponding to the intensity of crossover interference
`x`	vector of recombination rates
`inverse`	logical, if FALSE recombination rate is mapped to Morgan unit, if TRUE Morgan unit is mapped to recombination rate (default is FALSE)

Value

vector of genetic positions in Morgan units

References

Felsenstein, J. (1979) A mathematically tractable family of genetic mapping functions with different amounts of interference. Genetics 91:769-775.

Examples

```
felsenstein(0.1, seq(0, 0.5, 0.01))
```

geneticPosition *Estimation of genetic position*

Description

Estimation of genetic positions (in centi Morgan)

Usage

```
geneticPosition(final, map1, 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`
`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
`exclude`	optional vector (LEN < p) of SNP IDs to be excluded (e.g., candidates of misplaced SNPs; default NULL)
`threshold`	optional value; recombination rates <= threshold are considered for smoothing approach assuming theta ~ Morgan (default 0.05)

Details

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

The minimization problem `(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

list (LEN 2)

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

gen.Mb vector (LEN p) of physical positions of SNPs (in Mbp)

References

Qanbari, S. & Wittenburg, D. (2020) Male recombination map of the autosomal genome in German Holstein. Genetics Selection Evolution 52:73. doi: 10.1186/s1271102000593z

Examples

```
### 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)
### post-processing to achieve final and valid set of estimates
final <- editraw(res, map.chr)
```

```
### approximation of genetic positions
pos <- geneticPosition(final, map.chr)
```

genotype.chr *targetregion: progeny genotypes*

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 200 columns.

haldane *Haldane's genetic map function*

Description

Calculation of genetic distances from recombination rates

Usage

```
haldane(x, inverse = F)
```

Arguments

x	vector of recombination rates
inverse	logical, if FALSE recombination rate is mapped to Morgan unit, if TRUE Morgan unit is mapped to recombination rate (default is FALSE)

Value

vector of genetic positions in Morgan units

References

Haldane JBS (1919) The combination of linkage values, and the calculation of distances between the loci of linked factors. J Genet 8: 299-309.

Examples

```
haldane(seq(0, 0.5, 0.01))
```

`hapSire` *targetregion: sire haplotypes*

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 201 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, exclude = NULL, 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; 9 indicates missing genotype

`exclude` vector (LEN < p) of SNP IDs (for filtering column names of `genotype.chr`) to be excluded from analysis (default NULL)

`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` ID of 1. SNP

`SNP2` ID of 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) at `SNP1`

`p2` Maternal allele frequency (allele 1) at `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 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

Hampel, A., Teuscher, F., Gomez-Raya, L., Doschoris, M. & Wittenburg, D. (2018) Estimation of recombination rate and maternal linkage disequilibrium in half-sibs. Frontiers in Genetics 9:186. doi: 10.3389/fgene.2018.00186

Gomez-Raya, L. (2012) Maximum likelihood estimation of linkage disequilibrium in half-sib families. Genetics 191:195-213.

Examples

```
### 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)
### post-processing to achieve final and valid set of estimates
final <- editraw(res, map.chr)
```

`karlin` *Liberman and Karlin's genetic map function*

Description

Calculation of genetic distances from recombination rates given a parameter

Usage

```
karlin(N, x, inverse = F)
```

Arguments

`N` parameter (positive integer) required by the binomial model to assess the count (of crossover) distribution; `N = 1` corresponds to Morgan's map function

`x` vector of recombination rates

`inverse` logical, if FALSE recombination rate is mapped to Morgan unit, if TRUE Morgan unit is mapped to recombination rate (default is FALSE)

Value

vector of genetic positions in Morgan units

References

Liberman, U. & Karlin, S. (1984) Theoretical models of genetic map functions. Theor Popul Biol 25:331-346.

Examples

```
karlin(2, seq(0, 0.5, 0.01))
```

kosambi *Kosambi's genetic map function*

Description

Calculation of genetic distances from recombination rates

Usage

```
kosambi(x, inverse = F)
```

Arguments

x	vector of recombination rates
inverse	logical, if FALSE recombination rate is mapped to Morgan unit, if TRUE Morgan unit is mapped to recombination rate (default is FALSE)

Value

vector of genetic positions in Morgan units

References

Kosambi D.D. (1944) The estimation of map distance from recombination values. Ann. Eugen. 12: 172-175.

Examples

```
kosambi(seq(0, 0.5, 0.01))
```

LDHScpp *Expectation Maximisation (EM) algorithm*

Description

Expectation Maximisation (EM) algorithm

Usage

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

Arguments

`XGF1`	integer matrix of progeny genotypes in genomic family 1
`XGF2`	integer 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) at 1. SNP

`p2` Maternal allele frequency (allele 1) at 2. SNP

`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, exclude = NULL)
```

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 (0, 1, 2) on a single chromosome with p SNPs; 9 indicates missing genotype
`nmin`	scalar, minimum required number of progeny for proper imputation, default 30
`exclude`	vector (LEN < p) of SNP indices to be excluded from analysis

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

Ferdosi, M., Kinghorn, B., van der Werf, J., Lee, S. & Gondro, C. (2014) hsphase: an R package for pedigree reconstruction, detection of recombination events, phasing and imputation of half-sib family groups BMC Bioinformatics 15:172. https://CRAN.R-project.org/package=hsphase

Examples

```
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

```
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

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

Examples

```
data(targetregion)
hap <- makehaplist(daughterSire, hapSire)
```

makehappm *Make list of imputed haplotypes and estimate recombination rate*

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, exclude = NULL)
```

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 (0, 1, 2) on a single chromosome with p SNPs; 9 indicates missing genotype
`nmin`	scalar, minimum required number of progeny for proper imputation, default 30
`exclude`	vector (LEN < p) of SNP IDs (for filtering column names of `genotype.chr`) to be excluded from analysis

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
- **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
- **gen** vector (LEN p) of genetic positions of SNPs (in cM)

References

Ferdosi, M., Kinghorn, B., van der Werf, J., Lee, S. & Gondro, C. (2014) hsphase: an R package for pedigree reconstruction, detection of recombination events, phasing and imputation of half-sib family groups BMC Bioinformatics 15:172. https://CRAN.R-project.org/package=hsphase

Examples

```
 data(targetregion)
hap <- makehappm(unique(daughterSire), daughterSire, genotype.chr, exclude = paste0('V', 301:310))
```

map.chr *targetregion: physical map*

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 **markername** official SNP name

Format

An object of class `data.frame` with 200 rows and 6 columns.

rao *System of genetic-map functions*

Description

Calculation of genetic distances from recombination rates given a mixing parameter

Usage

```
rao(p, x, inverse = F)
```

Arguments

p	mixing parameter (see details); `0 <= p <= 1`
x	vector of recombination rates
inverse	logical, if FALSE recombination rate is mapped to Morgan unit, if TRUE Morgan unit is mapped to recombination rate (default is FALSE)

Details

Mixing parameter `p=0` would match to Morgan, `p=0.25` to Carter, `p=0.5` to Kosambi and `p=1` to Haldane map function. As an inverse of Rao's system of functions does not exist, NA will be produced if `inverse = T`. To approximate the inverse call function `rao.inv(p, x)`.

Value

vector of genetic positions in Morgan units

References

Rao, D.C., Morton, N.E., Lindsten, J., Hulten, M. & Yee, S (1977) A mapping function for man. Human Heredity 27: 99-104. doi: 10.1159/000152856

Examples

```
rao(0.25, seq(0, 0.5, 0.01))
```

rao inverse *Approximation to inverse of Rao's system of map functions*

Description

Calculation of recombination rates from genetic distances given a mixing parameter

Usage

```
rao.inv(p, x)
```

Arguments

p	mixing parameter (see details); `0 <= p <= 1`
x	vector in Morgan units

Details

Mixing parameter `p=0` would match to Morgan, `p=0.25` to Carter, `p=0.5` to Kosambi and `p=1` to Haldane map function.

Value

vector of recombination rates

References

Rao, D.C., Morton, N.E., Lindsten, J., Hulten, M. & Yee, S (1977) A mapping function for man. Human Heredity 27: 99-104. doi: 10.1159/000152856

Examples

```
rao.inv(0.25, seq(0, 01, 0.1))
```

---

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 (0, 1, 2) of genomic family with coupling phase sires (1) at SNP pair
`Fam2`	matrix (DIM n.progeny x 2) of progeny genotypes (0, 1, 2) 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

```
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)
```

`targetregion` *Description of the targetregion data set*

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

```
## 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 <- 301:500
## map information of target region
map.chr <- map[map$Chr == chr, ][window, ]
## matrix of sire haplotypes in target region
hapSire <- rlist::list.rbind(haps[[chr]])
sireID <- 1:length(unique(daughterSire))
hapSire <- cbind(rep(sireID, each = 2), hapSire[, window])
## matrix of progeny genotypes
genotype.chr <- X[, map.chr$SNP]
colnames(genotype.chr) <- map.chr$SNP
save(list = c('genotype.chr', 'hapSire', 'map.chr', 'daughterSire'),
      file = 'targetregion.RData', compress = 'xz')

## End(Not run)
```

Index

∗ **datasets**
- daughterSire, 5
- genotype.chr, 9
- hapSire, 10
- map.chr, 18

bestmapfun, 2

checkCandidates, 3
countNumbers, 5

daughterSire, 5

editraw, 5

felsenstein, 7

geneticPosition, 7
genotype.chr, 9

haldane, 9
hapSire, 10
hsrecombi, 10

karlin, 12
kosambi, 13

LDHScpp, 13
loglikfun, 14

makehap, 15
makehaplist, 16
makehappm, 17
map.chr, 18

rao, 18
rao inverse, 19
rao.inv *(*rao inverse*)*, 19

startvalue, 20

targetregion, 21