Package ‘hsphase’

February 20, 2015

Type Package
Title Phasing, Pedigree Reconstruction, Sire Imputation and
Recombination Events Identification of Half-sib Families Using
SNP Data
Version 2.0.1
Date 2014-6-7
Author Mohammad Ferdosi <mferdosi@myun.edu.au>, Cedric Gondro <cgondro@une.edu.au>
Maintainer Mohammad Ferdosi <mferdosi@myun.edu.au>
Depends snowfall, R (>= 3.1.0)
LinkingTo RcppArmadillo (>= 0.4.300.8.0), Rcpp (>= 0.11.2)
Imports Rcpp
Description Identification of recombination events, haplotype reconstruction, sire imputation
and pedigree reconstruction using half-sib family SNP data
License GPL (>= 3)
NeedsCompilation yes
Repository CRAN
Date/Publication 2015-01-10 17:04:44

R topics documented:

hsphase-package ................................................. 2
aio .............................................................. 3
bmh .............................................................. 5
c .............................................................. 6
cs ............................................................... 6
genotypes .......................................................... 7
hbp .............................................................. 8
hh .............................................................. 9
hss ........................................................... 10
imageplot ...................................................... 11
impute ......................................................... 12
hsphase-package

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>map</code></td>
<td></td>
</tr>
<tr>
<td><code>ohd</code></td>
<td></td>
</tr>
<tr>
<td><code>ohg</code></td>
<td></td>
</tr>
<tr>
<td><code>ohplot</code></td>
<td></td>
</tr>
<tr>
<td><code>para</code></td>
<td></td>
</tr>
<tr>
<td><code>pedigree</code></td>
<td></td>
</tr>
<tr>
<td><code>pedigreeNaming</code></td>
<td></td>
</tr>
<tr>
<td><code>phf</code></td>
<td></td>
</tr>
<tr>
<td><code>pm</code></td>
<td></td>
</tr>
<tr>
<td><code>pogc</code></td>
<td></td>
</tr>
<tr>
<td><code>readGenotype</code></td>
<td></td>
</tr>
<tr>
<td><code>recombinations</code></td>
<td></td>
</tr>
<tr>
<td><code>rplot</code></td>
<td></td>
</tr>
<tr>
<td><code>rpoh</code></td>
<td></td>
</tr>
<tr>
<td><code>ssp</code></td>
<td></td>
</tr>
</tbody>
</table>

Index 26

hsphase-package Phasing, Pedigree Reconstruction, Sire Imputation and Recombination Events Identification for Half-sib Families

Description

Identification of recombination events, haplotype reconstruction and sire imputation using half-sib family SNP data.

Details

| Package:   | hsphase |
| Type:      | Package |
| Version:   | 2.0.1   |
| Date:      | 2014-6-17 |
| License:   | GPL 3 |

Main Functions:

- `bmh`: Block partitioning
- `ssp`: Sire inference
- `aio`: Phasing
- `imageplot`: Image plot of the block structure
- `rpoh`: Reconstruct pedigree based on opposing homozygote

Auxiliary Functions:

- `hss`: Half-sib family splitter
- `cs`: Chromosome splitter
**para:** Parallel data analysis

**Note:** These functions can be used to analyse large datasets.

**Author(s)**
Mohammad H. Ferdosi <mferdosi@une.edu.au>, Cedric Gondro <cgondro2@une.edu.au> Maintainer: Mohammad H. Ferdosi <mferdosi@myune.udu.au>

**References**


**Examples**
```r
genotype <- matrix(c( 0,0,0,0,1,2,2,2,0,0,2,0,0,0, 2,2,2,2,1,0,0,0,2,2,2,2,2, 2,2,2,2,1,2,2,2,0,2,2,2,2, 2,2,2,2,0,0,0,2,2,2,2,2, 0,0,0,0,2,2,2,2,0,0,0,0), ncol = 14, byrow = TRUE)
ssp(bmh(genotype), genotype)
aio(genotype)
imageplot(bmh(genotype), title = "ImagePlot example")
rplot(genotype, c(1:14))
```

---

**aio**

**All-in-one Phasing**

**Description**
Phasing of a half-sib family group.
Usage

aio(genotypeMatrix, bmh_forwardVectorSize = 30, bmh_excludeFP = TRUE,
    bmh_nsap = 3, output = "phase")

Arguments

- **genotypeMatrix**: matrix half-sib genotypes (one half-sib per row, with SNP ordered by mapping position in the columns. Data should be numeric. Use 0, 1 and 2 respectively for AA, AB and BB. Use 9 for missing data)
- **bmh_forwardVectorSize**: integer number of heterozygous sites used to validate recombination events or check for genotyping errors
- **bmh_excludeFP**: logical excludes SNPs that may cause heterozygous sites in the sire due to genotyping errors or map errors
- **bmh_nsap**: integer number of SNP per block to validate recombinations
- **output**: character if equal to the phase the 'aio' will only return the phasing results

Details

This function calls the `bmh`, `ssp` and `phf` functions.

Value

Returns a list of matrices. The first element (phasedHalfsibs) is a matrix with two rows (phased haplotypes) per individual (first paternal and second maternal). Data in format 0 (A), 1 (B) and 9 (unphased or missing). The second (sireHaplotype) and third (blockStructure) elements are the same as the output of `ssp` and `bmh`.

Note

Only this function needs to be called to phase a half-sib family. The genotype’s matrix must contain individuals from only one half-sib family and one ordered chromosome.

See Also

`bmh`, `ssp` and `phf`

Examples

```r
genotype <- matrix(c(  
  2,1,0,  
  2,0,0,  
  0,0,2  
), byrow = TRUE, ncol = 3)  
# There are 3 individuals with three SNPs
aio(genotype)  
# The genotypes must include only one half-sib family and one chromosome
```
**bmh**

**Block Partitioning**

**Description**

Identifies the block structure (chromosome segments) in the half-sib family that each individual inherited from its sire.

**Usage**

```r
bmh(GenotypeMatrix, forwardVectorSize = 30, excludeFP = TRUE, nsap = 3)
```

**Arguments**

- `GenotypeMatrix` matrix half-sib genotypes (one half-sib per row, with SNP ordered by mapping position in the columns. Data should be numeric. Use 0, 1 and 2 respectively for AA, AB and BB. Use 9 for missing data)
- `forwardVectorSize` integer number of heterozygous sites used to validate recombination events or check for genotyping errors (50k -> 30, 700k -> 120)
- `excludeFP` logical excludes SNPs that may cause heterozygous sites in the sire due to genotyping errors or map errors
- `nsap` integer number of SNP per block to validate recombinations (50k -> 3, 700k -> 10)

**Value**

Returns a matrix of the blocking structure that contains 1s, 2s and 0s. 1s and 2s are the two sire strands. The choice of strand is arbitrary for each chromosome and not consistent across chromosomes. 0s indicate regions of unknown origin.

**Note**

The genotype’s matrix must contain individuals from only one half-sib family and one ordered chromosome.

**See Also**

`ssp`, `phf`, `aio` and `imageplot`

**Examples**

```r
genotype <- matrix(c(0,2,1,1,1, 2,0,1,2,2, 2,2,1,0,2, 2,2,1,1,1),
```

```r
5 3
```
Description
Detect all possible crossover events.

Usage
co(genotypeMatrix)

Arguments
genotypeMatrix matrix half-sib genotypes (one half-sib per row, with SNP ordered by mapping position in the columns. Data should be numeric. Use 0, 1 and 2 for respectively AA, AB and BB. Use 9 for missing data)

Value
Returns a matrix with the number of crossover events for each site.

Examples

geno <- matrix(c(2,1,0,2,0,2,0,0,2), byrow = TRUE, ncol = 3) # Define a Half-sib Genotype Matrix
# Individual 1
# Individual 2
# Individual 3
# There are 3 individuals with three SNPs
co(geno)

Description
This function splits the genotypes list generated by hss into the different chromosomes based on a map file and orders SNP based on chromosomal position.

Usage
cs(halfsib, mapPath, separator = " ")
Arguments

halfsib: list list with matrices of half-sib genotypes, one family per list item

mapPath: character path to the map file (column 1 -> SNP names, column 2 -> chromosome name and column 3 -> SNP position in base pairs) or, alternatively, the name of a dataframe with the mapping information (in the same format)

separator: character separator character used in the map file

Details

The map file should include only the chromosomes that will be analyzed. For example, the Y and X chromosomes should be excluded (and others optionally). Names of each element in the list can be used for further categorization. The header must be "Name Chr Position".

Value

Returns a list of matrices, the number of elements in this list is the number of half-sib families multiplied by the number of chromosomes.

Examples

# Please run demo(hsphase)

---

<table>
<thead>
<tr>
<th>genotypes</th>
<th>Example of Genotype Data Set</th>
</tr>
</thead>
</table>

Description

An example of genotype matrix for hsphase.

Usage

data(genotypes)

Format

A genotype matrix with the following columns and rows:

- Columns: SNPs
- Rows: Animals
Description

Creates a blocking structure matrix of the half-sib family based on phased data of the sire and half-sib family.

Usage

hbp(PhasedGenotypeMatrix, PhasedSireGenotype, strand = "auto")

Arguments

PhasedGenotypeMatrix
  matrix haplotypes for a half-sib family (two rows per individual)
PhasedSireGenotype
  matrix haplotypes of sire
strand
  character method for identification of paternal strand (1 and 2 for strand one and two of the offsprings)

Value

Returns a matrix where 3 or 4 stands for the SNP originating in, respectively, strands 1 and 2. 0 indicates that the source strand for the SNP is unknown.

Note

The input matrices must only contain individuals from one half-sib family and one ordered chromosome. The strand option should be set to "auto" (default value).

See Also

aio, ssp

Examples

sire <- matrix(c(
  0,0,0,0,1,
  0,1,1,1,0
), byrow = TRUE, ncol = 6) # Haplotype one of the sire
haplotypeHalfsib <- matrix(c(
  1,0,1,1,1,1,
  0,1,0,0,0,0,
  0,1,1,0,1,1,
  1,0,0,1,0,0
), byrow = TRUE, ncol = 6) # Individual one, haplotype one
# Individual one, haplotype two
# Individual two, haplotype one
# Individual two, haplotype two
# 0s and 1s are alleles a and b
**hh**

hbp(haplotypeHalfsib, sire)

---

**Heatmap of Half-sibs**

---

**Description**

The `hh` function creates a heatmap of the half-sib families using the matrix of opposing homozygotes.

**Usage**

```
hh(oh, inferredPedigree, realPedigree, pedOnly = TRUE)
```

**Arguments**

- `oh` matrix Opposing homozygotes matrix (output of `ohg`)
- `inferredPedigree` matrix inferred pedigree (output of `rpoh`)
- `realPedigree` matrix original pedigree
- `pedOnly` logical Consider only individuals that are exist in the real pedigree

**Value**

Returns the heatmap of the matrix of opposing homozygotes with sidebars colour coded by sires from the inferred and original pedigree.

**Author(s)**

The function uses the colour generated by `getcol` function in the `made4` package (Aedin Culhane).

**See Also**

- `ohg` and `rpoh`

**Examples**

```
c1h1 <- simulateHalfsib(numInd = 62, numSNP = 5000)
c1h2 <- simulateHalfsib(numInd = 38, numSNP = 5000)
Genotype <- rbind(c1h1, c1h2)
oh <- ohg(Genotype) # creating the Opposing Homozygote matrix
hh(oh)
```
**Half-sib Family Splitter**

**Description**

Splits the dataset into half-sib family groups based on a pedigree.

**Usage**

```r
hss(pedigree, genotype, check = TRUE)
```

**Arguments**

- `pedigree`: matrix the pedigree matrix should contain at least two columns, the first column with the half-sib IDs and the second column with the sires IDs
- `genotype`: matrix genotype matrix with SNP ordered by mapping position in the columns. Data should be numeric. Use 0, 1 and 2 respectively for AA, AB and BB. Use 9 for missing data
- `check`: logical check the genotype file for the possible errors

**Details**

Only half-sib groups that have more than 3 individuals will be returned.

**Value**

Returns a list of numeric matrices, each matrix is a half-sib family.

**Note**

Pedigree must have at least two columns with sample ids (Column 1) and sire ids (Column 2).

**Examples**

```r
# Please run demo(hsphase)
```
imageplot  
Image Plot of Blocking Structure

Description
Create an imageplot of the blocking structure.

Usage
imageplot(x, title, rv = FALSE, ...)

Arguments
- x: matrix blocking structure (output of \texttt{bmh} or \texttt{hbp} functions)
- title: character title of imageplot
- rv: logical reverse the colour
- ...

Details
White indicates regions of unknown origin, red and blue correspond to the two sire strands.

Author(s)
This is a modified version of a function written by Chris Seidel.
http://www.phaget4.org/R/image_matrix.html

See Also
\texttt{bmh} and \texttt{aio}

Examples
```r
genotype <- matrix(c(0,2,1,1,1,2,0,1,2,2,2,2,1,0,2,2,2,1,1,1,0,0,2,1,0), ncol = 5, byrow = TRUE) # each row contains the SNP of individuals
text("imageplot(bmh(genotype))")
```
impute

Impute of Low Density SNP Marker to High Density (Paternal Strand)

Description

Impute the paternal strand from low density to high density utilising high density sire haplotype.

Usage

```r
impute(halfsib_genotype_ld, sire_hd, bmh_forwardVectorSize = 30,
       bmh_excludefp = TRUE, bmh_nsap = 3)
```

Arguments

- `halfsib_genotype_ld` : matrix half-sib genotypes with low density marker (one half-sib per row, with SNP ordered by mapping position in the columns. Data should be numeric. Use 0, 1 and 2 respectively for AA, AB and BB. Use 9 for missing data)
- `sire_hd` : matrix haplotype of sire (this parameter can be sequence data or any phased sire - the matrix should have rownames which are the sample IDs and colnames which are the SNP names)
- `bmh_forwardVectorSize` : integer number of heterozygous sites used to validate recombination events or check for genotyping errors
- `bmh_excludefp` : logical exclude SNPs that may cause heterozygous sites in the sire due to genotyping errors or map errors
- `bmh_nsap` : integer number of SNPs per block

Value

Return an imputed half-sib matrix.

See Also

`bmh`, `ssp` and `phf`
Example of Map File

Description
An example of map for hphase.

Usage
```
data(map)
```

Format
A data frame with the following columns:
- Name: SNP name
- Chr: chromosome
- Position: SNP position in base pairs

Opposing Homozygote Detection

Description
Counts the number of opposing homozygotes for each animal that caused a heterozygous site in the sire.

Usage
```
ohd(genotypeMatrix, unique_check = FALSE, SNPs = 6000)
```

Arguments
- `genotypeMatrix`: matrix half-sib genotypes (one half-sib per row, with SNP ordered by mapping position in the columns. Data should be numeric. Use 0, 1 and 2 respectively for AA, AB and BB. Use 9 for missing data)
- `unique_check`: logical check if samples uniquely originate an opposing homozygote at a locus
- `SNPs`: integer number of SNP to use

Value
Returns a vector with the number of heterozygous sites that each sample caused.

Note
This function can be used to identify pedigree errors; i.e., the outliers.
Author(s)

This method is suggested by Bruce Tier <btier@une.edu.au> to identify pedigree errors.

Examples

```r
genotype <- matrix(c(  
  2,1,0,  
  2,0,0,  
  0,0,2  
  ), byrow = TRUE, ncol = 3)

ohd(genotype)
```

---

**Ohg**

*Matrix of Opposing Homozygotes*

**Description**

Creates a matrix of opposing homozygotes from the genotype matrix.

**Usage**

```r
ohg(genotypeMatrix)
```

**Arguments**

- `genotypeMatrix` (matrix) genotype (Data should be numeric. Use 0, 1 and 2 respectively for AA, AB and BB. Use 9 for missing data)

**Value**

Returns a square matrix (sample X sample) with the pairwise counts of opposing homozygotes.

**Note**

This function can be slow with a large data set. The fast version of this function will be available after publish of the related manuscript.

**Author(s)**


**See Also**

`rpolh`
ohplot

Examples

genotype <- matrix(c(  
    2,1,0,  
    2,0,0,  
    0,0,2  
  ), byrow = TRUE, ncol = 3)

  ohg(genotype)

---

Opposing Homozygotes Plot

Description

Plot the sorted vectorized matrix of Opposing Homozygotes.

Usage

ohplot(oh, genotype, pedigree, check = FALSE)

Arguments

- oh: integer Opposing homozygotes matrix (Output of ohg)
- genotype: matrix genotype of one chromosome (data should be numeric. Use 0, 1 and 2 for respectively AA, AB and BB. Use 9 for missing data)
- pedigree: matrix the pedigree matrix should contain at least two columns, the first column with the half-sib IDs and the second column with the sires IDs. This argument is optional.
- check: logical check the genotype file for the possible errors

Details

The cut off line shows the edge of most different groups.

See Also

ohg and rpol

Examples

set.seed(100)
chr <- list()
sire <- list()
set.seed(1)
chr <- list()
for(i in 1:5)  
{  
  chr[[i]] <- .simulateHalfsib(numInd = 20, numSNP = 5000, recbound = 1:10)
sire[[i]] <- ssp(bmh(chr[[i]]), chr[[i]])
sire[[i]] <- sire[[i]][1,] + sire[[i]][2,]
sire[[i]][sire[[i]] == 18] <- 9

Genotype <- do.call(rbind, chr)
rownames(Genotype) <- 6:(nrow(Genotype) + 5)
sire <- do.call(rbind, sire)
rownames(sire) <- 1:5
Genotype <- rbind(sire, Genotype)
oh <- ohg(Genotype)  # creating the Opposing Homozygote matrix
pedigree <- as.matrix(data.frame(c(1:5, 6:(nrow(Genotype))),
rep = c(rep(0,5), rep(1:5, rep(20,5)))))
ohplot(oh, Genotype, pedigree, check = TRUE)

para        Parallel Analysis of Data

Description

This function uses the list of matrices (the output of `cs`) and runs one of the options, on each element of the list, in parallel.

Usage

```r
para(halfsibs, cpus = 1, option = "bmh", type = "SOCK", bmh_forwardVectorSize = 30,
bmh_excludeFP = TRUE, bmh_nsap = 3, pmMethod = "constant")
```

Arguments

- `halfsibs`: list list of matrices of half-sibs (can be generated with `hss` and `cs` functions)
- `cpus`: numeric number of CPUs (thread)
- `option`: character type of analysis
- `type`: character type of cluster for parallel analysis
- `bmh_forwardVectorSize`: integer number of heterozygous sites used to validate recombination events or check for genotyping errors
- `bmh_excludeFP`: logical exclude SNPs that may cause heterozygous sites in the sire due to genotyping errors or map errors
- `bmh_nsap`: integer number of SNPs per block
- `pmMethod`: character method for creating the recombination matrix

Details

Type of analysis can be `bmh`, `ssp`, `aio`, `pm`, or `rec` (refer to `pm`, `rplot` and vignette for more information about `rec`).
**pedigree**

**Value**

Returns a list of matrices with the results (formats specific to the option selected).

**Examples**

```r
# Please run demo(hsphase)
```

---

**pedigree**  

<table>
<thead>
<tr>
<th>Example Pedigree</th>
</tr>
</thead>
<tbody>
<tr>
<td>pedigree</td>
</tr>
</tbody>
</table>

---

**Description**

An example pedigree for hsphase.

**Usage**

```r
data(pedigree)
```

**Format**

a data frame with the following columns:

- First Column: half-sibs
- Second Column: sires

---

**pedigreeNaming**  

<table>
<thead>
<tr>
<th>Fix Pedigree Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>pedigreeNaming</td>
</tr>
</tbody>
</table>

---

**Description**

Tries to link the inferred pedigree from `rpoh` with the sire IDs in the original pedigree and fix pedigree errors.

**Usage**

```r
pedigreeNaming(inferredPedigree, realPedigree)
```

**Arguments**

- `inferredPedigree`  
  matrix inferred pedigree (output of `rpoh`)
- `realPedigree`  
  matrix original pedigree
### Details
This function calls the bmh and recombinations functions to count the number of recombinations in each half-sib group.

### Value
Returns the inferred pedigree with the best fit to the sire names used in the original pedigree file.

### See Also
- rpoh and ohg

### Examples
```r
# Please run demo(hsphase)
```

---

**phf** 
*Half-Sib Family Phasing*

### Description
Phases the half-sib family by using the blocking structure and imputed sire matrices.

### Usage
```r
phf(GenotypeMatrix, blockMatrix, sirePhasedMatrix)
```

### Arguments
- **GenotypeMatrix** matrix half-sib genotypes (one half-sib per row, with SNP ordered by mapping position in the columns. Data should be numeric. Use 0, 1 and 2 respectively for AA, AB and BB. Use 9 for missing data)
- **blockMatrix** matrix blocking structure (output of bmh)
- **sirePhasedMatrix** matrix imputed sire (output of ssp)

### Value
Returns a matrix that contains the phased parental haplotypes of the half-sibs. It uses 1, 0 and 9 for A, B and missing.

### Note
The genotype matrix must only contain individuals from one half-sib family and one ordered chromosome. This function is used by the aio function for complete phasing of a half-sib group.
See Also
aio

Examples

```r
genotype <- matrix(c(
  2,1,0,
  2,0,0,
  0,0,2), byrow = TRUE, ncol = 3)
block <- bmh(genotype)
phf(genotype, block, ssp(block, genotype))
```

---

**Description**

Creates a recombination matrix based on the blocking structure.

**Usage**

```r
pm(blockMatrix, method = "constant")
```

**Arguments**

- `blockMatrix` matrix blocking structure (Output of bmh)
- `method` character method for creating the recombination matrix

**Details**

This function finds the recombination between two consecutive sites, and marks the recombination site with a 1; if there are unknown sites between two blocks it will also mark these sites with a 1 (constant method) or 1 divided by number of unknown site (relative method).

**Examples**

```r
genotype <- matrix(c(
  0,2,0,1,0,
  2,0,1,2,2,
  2,2,1,0,2,
  2,2,1,1,1,
  0,0,2,1,0), ncol = 5, byrow = TRUE) # each row contains the SNP of individuals

(result <- bmh(genotype))
pm(result)
```
pogc  

_Parent Offspring Group Constructor_

**Description**

Assign offsprings to the parents.

**Usage**

```
pogc(oh, genotypeError)
```

**Arguments**

- `oh` integer opposing homozygotes matrix (Output of `ohg`)
- `genotypeError` integer number of genotypeing error allowed in the `oh` matrix

**Value**

Return a data frame with two columns. The first column is the animal ID and the second column is the parent ID.

**See Also**

`ohg`, `hss` and `rpoh`

**Examples**

```r
set.seed(100)
chr <- list()
sire <- list()
set.seed(1)
chr <- list()
for(i in 1:5)
{
  chr[[i]] <- .simulateHalfsib(numInd = 20, numSNP = 5000, recbound = 1:10)
sire[[i]] <- ssp(bmh(chr[[i]]), chr[[i]])
sire[[i]][sire[[i]][1,] + sire[[i]][2,] == 18] <- 9
}
Genotype <- do.call(rbind, chr)
rownames(Genotype) <- 6:(nrow(Genotype) + 5)
sire <- do.call(rbind, sire)
rownames(sire) <- 1:5
Genotype <- rbind(sire, Genotype)
oh <- ohg(Genotype)  # creating the Opposing Homozygote matrix
pogc(oh, 5)
```
readGenotype

Description

This function reads and checks genotype files.

Usage

readGenotype(genotypePath, separatorGenotype = " ", check = TRUE)

Arguments

genotypePath character genotype path (animals (rows) and SNP (columns), SNP should be coded as 0, 1 and 2 for respectively AA, AB and BB. Use 9 for missing data. please refer to vignette for more information)

separatorGenotype character separator character for genotype

check logical check the genotype file for possible errors

Value

Returns the genotype matrix.

Note

Please refer to vignette for more information.

Examples

# A comprehensive demo and example dataset is available from

recombinations

Description

Counts the number of recombinations for each individual.

Usage

recombinations(blockMatrix)
Arguments

- `blockMatrix` matrix block structure (Output of `bmh`)

Value

Returns a vector of recombinations. The number of elements in this vector is equal to the number of individuals, i.e. each element holds the number of recombinations identified for each sample.

See Also

`bmh`

Examples

```r
genotype <- matrix(c(
  2,1,0,0,
  2,0,2,2,
  0,0,2,2,
  0,2,0,0
), byrow = TRUE, ncol = 4)

recombinations(bmh(genotype))
```

---

**rplot**

**Recombination Plot**

Description

This function creates a plot which shows the sum of all recombination events across a half-sib family.

Usage

```r
rplot(x, distance, start = 1, end = ncol(x), maximum = 100, overwrite = FALSE, method = "constant")
```

Arguments

- `x` matrix of half-sib genotypes (one half-sib per row, with SNP ordered by mapping position in the columns. Data should be numeric. Use 0, 1 and 2 for respectively AA, AB and BB. Use 9 for missing data).
- `distance` integer of physical distances between markers
- `start` integer first marker selected for the plot
- `end` integer last marker selected for the plot
- `maximum` integer maximum number of recombinations to show (higher recombination rates will be omitted from the plot)
- `overwrite` logical draw a diagram over the current diagram (default FALSE)
- `method` character please refer to the *pm* document
**Examples**

```r
geno <- matrix(c(0,2,0,1,0,
                 2,0,1,2,2,
                 2,2,1,0,2,
                 2,2,1,1,1,
                 0,0,2,1,0), ncol = 5, byrow = TRUE) # each row contains the SNP of individuals

rplot(geno, c(1,2,3,4,8))
```

---

**Description**

Reconstructs a half-sib pedigree based on a matrix of opposing homozygotes.

**Usage**

```r
rpoh(genoMatrix, oh, forwardVectorSize = 30, excludeFP = TRUE, nsap = 3,
     maxRec = 15, intercept = 26.3415, coefficient = 77.3171, snpnooh, method, maxsnpnooh)
```

**Arguments**

- `genotypeMatrix`: matrix genotype of one chromosome (data should be numeric. Use 0, 1 and 2 for respectively AA, AB and BB. Use 9 for missing data)
- `oh`: integer Opposing homozygotes matrix (Output of `ohg`)
- `forwardVectorSize`: integer number of heterozygous sites used to validate recombination events or check for genotyping errors
- `excludeFP`: logical excludes SNPs that may cause heterozygous sites in the sire due to genotyping errors or map errors
- `nsap`: integer number of SNP per block to validate recombinations
- `maxRec`: integer maximum number of expected recombinations per individual
- `intercept`: integer intercept of fitted model
- `coefficient`: integer coefficient of fitted model
- `snpnooh`: integer number of SNPs used to create `oh` matrix (this number must be divided by 1000)
- `method`: character pedigree reconstruction method
- `maxsnpnooh`: numeric the maximum number of allowing opposing homozygote in a half-sib family
Details

Four methods simple, recombinations, calus and manual can be utilized to reconstruct the pedigree.

The following examples show the arguments require for each method.

```r
pedigree1 <- rpoh(oh = oh, snpnooh = 732, method = "simple")
pedigree2 <- rpoh(genotypeMatrix = genotypeChr1, oh = ohg(genotype), maxRec = 10, method = "recombinations")
pedigree3 <- rpoh(genotypeMatrix = genotype, oh = oh, method = "calus")
pedigree4 <- rpoh(oh = oh, maxsnpnooh = 31662, method = "manual")
```

Value

Returns a data frame with two columns, the first column is animals’ ID and the second column is sire identifiers (randomly generated).

Note

Method can be recombinations, simple, calus or manual. Please refer to vignette for more information.

The sire genotype should be removed before using this function utilizing pogc function.

See Also

bhm and recombinations

Examples

```r
# Please run demo(hsphase)
```

---

ssp

---

Sire Imputation and Phasing

Description

Infer (impute) and phase sire’s genotype based on the block structure matrix (recombination blocks) and homozygous sites of the half-sib genotype matrix.

Usage

```r
ssp(blockMatrix, genotypeMatrix)
```

Arguments

- `blockMatrix`: matrix block structure (Output of bhm)
- `genotypeMatrix`: matrix half-sibs genotype (each row includes the SNP of individuals, 0, 1 and 2 for respectively AA, AB and BB. Use 9 for missing data)
ssp

Value
Returns a matrix (Imputed Sire) with two rows one for each haplotype of the sire (columns are SNP in the order of the genotype matrix). Alleles are coded as 0 (A) and 1 (B). Alleles that could not be imputed are coded as 9.

See Also
phf, aio and imageplot

Examples
```r
genotype <- matrix(c( 0,2,1,1,1, 2,0,1,2,2, 2,2,1,0,2, 2,2,1,1,1, 0,0,2,1,0), ncol = 5, byrow = TRUE) # each row contains the SNP of individuals

(result <- ssp(bmh(genotype), genotype))
```
Index

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chromosome</td>
<td>cs</td>
</tr>
<tr>
<td>High_Density</td>
<td>impu</td>
</tr>
<tr>
<td>Low_Density</td>
<td>impu</td>
</tr>
<tr>
<td>Opposing_Homozygotes</td>
<td>ohpl</td>
</tr>
<tr>
<td>SNP</td>
<td>hbp</td>
</tr>
<tr>
<td>Splitter</td>
<td>cs</td>
</tr>
<tr>
<td>block, half-sib, sire</td>
<td></td>
</tr>
<tr>
<td>inference, phase</td>
<td></td>
</tr>
<tr>
<td>block</td>
<td>hbp</td>
</tr>
<tr>
<td>datasets</td>
<td></td>
</tr>
<tr>
<td>genotypes</td>
<td></td>
</tr>
<tr>
<td>map</td>
<td></td>
</tr>
<tr>
<td>pedigree</td>
<td></td>
</tr>
<tr>
<td>crossover</td>
<td>co</td>
</tr>
<tr>
<td>error</td>
<td></td>
</tr>
<tr>
<td>block</td>
<td></td>
</tr>
<tr>
<td>crossover</td>
<td>co</td>
</tr>
<tr>
<td>datasets</td>
<td></td>
</tr>
<tr>
<td>genotypes</td>
<td></td>
</tr>
<tr>
<td>map</td>
<td></td>
</tr>
<tr>
<td>pedigree</td>
<td></td>
</tr>
<tr>
<td>error</td>
<td></td>
</tr>
<tr>
<td>datasets</td>
<td></td>
</tr>
<tr>
<td>genotypes</td>
<td></td>
</tr>
<tr>
<td>error</td>
<td></td>
</tr>
<tr>
<td>genotype</td>
<td>ogh</td>
</tr>
<tr>
<td>haplotype</td>
<td></td>
</tr>
<tr>
<td>aio</td>
<td></td>
</tr>
<tr>
<td>phf</td>
<td></td>
</tr>
<tr>
<td>heatmap</td>
<td>hh</td>
</tr>
<tr>
<td>image</td>
<td></td>
</tr>
<tr>
<td>impute</td>
<td></td>
</tr>
<tr>
<td>inference</td>
<td></td>
</tr>
<tr>
<td>opposing homozygote</td>
<td>ohd</td>
</tr>
<tr>
<td>opposing-homozygote</td>
<td>ogh</td>
</tr>
<tr>
<td>parentage</td>
<td></td>
</tr>
<tr>
<td>pedigree</td>
<td></td>
</tr>
<tr>
<td>pedigreenaming</td>
<td></td>
</tr>
<tr>
<td>parentage</td>
<td></td>
</tr>
<tr>
<td>plot</td>
<td></td>
</tr>
<tr>
<td>reconstruction</td>
<td></td>
</tr>
<tr>
<td>snp</td>
<td></td>
</tr>
</tbody>
</table>
INDEX

aio, 3
bmh, 5
ohg, 14
phf, 18
rplot, 22
rpoh, 23
ssp, 24

aio, 2, 3, 5, 8, 11, 16, 18, 19, 25

bmh, 2, 4, 5, 11, 12, 16, 18, 19, 22, 24

go, 6
cs, 2, 6, 16

genotypes, 7

hbp, 8, 11
hh, 9
hsphase (hsphase-package), 2
hsphase-package, 2
hss, 2, 6, 10, 16, 20

imageplot, 2, 5, 11, 25
impute, 12

map, 13

ohd, 13
ohg, 9, 14, 15, 18, 20, 23
ohplot, 15

para, 3, 16
pedigree, 17
pedigreeNaming, 17
phf, 4, 5, 12, 18, 25
pm, 16, 19, 22
pogc, 20, 24

readGenotype, 21
recombinations, 18, 21, 24
rplot, 16, 22
rpoh, 2, 9, 14, 15, 17, 18, 20, 23

ssp, 2, 4, 5, 8, 12, 16, 18, 24