Package 'PolyHaplotyper'

June 17, 2021

Type Package

Title Assignment of Haplotypes Based on SNP Dosages in Diploids and Polyploids

Version 1.0.1

Author Roeland E. Voorrips

Maintainer Roeland E. Voorrips <roeland.voorrips@wur.nl>

Description Infer the genetic composition of individuals in terms of haplotype dosages for a haploblock, based on bi-allelic marker dosages, for any ploidy level.
Reference: Voorrips and Tumino: PolyHaplotyper: haplotyping in polyploids based on bi-allelic marker dosage data.
Submitted to BMC Bioinformatics (2021).

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports XML

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2021-06-17 14:20:05 UTC

R topics documented:

allhap 2
allHaplotypes 3
build_ahccompletelist 4
calcMrkHaptable 5
calcStatistics 6
checkmrkDosage 7

compareHapMrkDosages . . . 8
compareHapresults . . . 9
demo_ped . . . 10
demo_snpdos . . . 10
expandHapdos . . . 10
getFSfreqs . . . 11
getGameteFreqs . . . 12
hapcomb2hapdos . . . 13
hapdos2hapcomb . . . 14
hapdos2mrkdos . . . 15
haploblock_df2list . . . 16
inferHaplotypes . . . 17
make.Happyinf.input . . . 20
make.SATlotyper.input . . . 21
make.ShesisPlus.input . . . 22
mergeReplicates . . . 23
mrkdid2mrkdos . . . 24
mrkdos2mrkdid . . . 25
overviewByFS . . . 26
padded . . . 27
pedigreeHapCheck . . . 28
pedigreeSim2PH . . . 29
phblocks . . . 30
phdos . . . 31
phFS . . . 31
phpar . . . 31
phped . . . 32
phresults . . . 32
read.Happyinf.output . . . 32
read.SATlotyper.output . . . 33
read.ShesisPlus.output . . . 34
run.SATlotyper . . . 35
showOneFS . . . 36
totHapcombCount . . . 37
usedhap . . . 38

Index **39**

`allhap` *Find all possible haplotypes*

Description

Find all possible haplotypes for a haploblock from the haplotyping result

Usage

```
allhap(hapresults, haploblock)
```

Arguments

hapresults	list as returned by inferhaplotypes, or one element of such a list (i.e. the results for one haploblock)
haploblock	if hapresults is one element of the return value of inferHaplotypes, haploblock should be missing of NULL; else haploblock is a single value indicating the haploblock: either its name of its index in hapresults

Details

This function works with the results of inferHaplotypes; the setting of dropUnused does not affect this function

Value

an array with all possible haplotypes. The haplotypes are in columns, with the haplotype numbers as colnames; the markers are in rows.

Examples

```
data(PolyHaplotyper_small)
# show the composition of all possible haplotypes with the markers
# in the first haploblock:
allhap(hapresults=phresults, haploblock=1)
```

---

allHaplotypes *get all haplotypes for the given markers*

Description

Given a set of bi-allelic (SNP) marker names, generate all possible haplotypes

Usage

```
allHaplotypes(mrknames)
```

Arguments

mrknames	the names of the (bi-allelic) markers in the haploblock (contig)

Value

a matrix with markers in columns and all possible (2 ^ nmrk) haplotypes in rows. 0: haplotype contains the non-dosage-counted marker allele (the reference allele); 1: haplotype contains the dosage-counted (alternative) marker allele. The colnames are the marker names.

Examples

```
# show the 8 possible haplotypes with 3 bi-allelic markers:
allHaplotypes(mrknames=c("mrkA", "mrkB", "mrkC"))
```

`build_ahccompletelist` *generate a list with all haplotype combinations*

Description

generate a list which contains for each marker dosage combination at a given ploidy all matching haplotype combinations

Usage

```
build_ahccompletelist(ploidy, maxmrk, savesec=1800, printsec=300,
overwrite, shorten=FALSE)
```

Arguments

`ploidy`	ploidy; may be even or odd, but inferHaplotypes only works with even ploidy if FS families are present, so building an ahccompletelist for odd ploidy is probably not useful
`maxmrk`	the list will countain all haplotype combinations for haploblocks of 1...maxmrk markers
`savesec`	default 1800: number of seconds between successive saves of the intermediate results
`printsec`	default 300: number of seconds between printout of the current set of haplotypes. NA or 0 suppresses printing.
`overwrite`	the new file ahccompletelist_<ploidy>x.RData is written in the working directory. If a file with that name already exists and would be changed, this is aborted and the old file maintained if overwrite is FALSE
`shorten`	if file ahccompletelist_<ploidy>x.RData already exists and covers haploblocks of lengths longer than maxmrk, this file is left unchanged if shorten is FALSE (default), but is shortened to maxmrk if shorten is TRUE

Details

An ahccompletelist reduces the processing time of inferHaplotypes enormously but takes a long time to build. This should therefore be done when PolyHaplotyper will be used multiple times.
If an ahccompletelist file already exists in the working directory, this file is used as starting point; if maxmrk is larger than the length of the existing list only the additional items are calculated.
If this function crashes (due to exceeding the memory limits of the computer or of R) the file ahccompletelist_<ploidy>x.RData in the working directory contains an intact version of the ahccompletelist for the last finished marker count. Also, the temporary file "buildAHCcompletelist_<datetime>"_"<nmrk>.RData" contains a part of the list item for the number of markers being calculated at that moment.
Note that this function takes lots of time and memory already for ploidy 4 / maxmrk 7 and ploidy 6 / maxmrk 5, and is not practicable above ploidy 4 / maxmrk 8, ploidy 6 / maxmrk 6, ploidy 8 / maxmrk 5. Also, the files are large and take up considerable memory, and most of the size is used for the largest haploblock size. Therefore maxmrk should not be taken larger than necessary.

Value

NULL (invisible); the actual result is a file in the working directory named ahccompletelist_<ploidy>x.RData

Examples

```
# this example will create a file ahccompletelist_4x.RData
# in the working directory, for ploidy=4x and 1 - 4 markers:
build_ahccompletelist(ploidy=4, maxmrk=4, overwrite=FALSE)
```

---

calcMrkHaptable *produce a table of nr of markers vs nr of haplotypes*

Description

produce a table of nr of markers vs nr of haplotypes

Usage

```
calcMrkHaptable(ovwFS)
```

Arguments

ovwFS	a list as produced by overviewFS. Only the first 2 columns of list item ovw are used

Value

a frequency table with the numbers of haploblocks, with all combinations of marker counts and inferred haplotype counts per haploblock. The column with haplotype count NA (if any) shows the haploblocks for which no haplotype solution was found (the reason for that would usually be found in column 1 of ovwFS$messages)

Examples

```
data(PolyHaplotyper_small)
phovw <- overviewByFS(haploblock=phblocks, parents=phpar, FS=phFS,
                      hapresults=phresults)
# in this small dataset there are only 2 haploblocks, each with 4 markers:
calcMrkHaptable(ovwFS=phovw)
# in both haploblocks 5 haplotypes are inferred
```

`calcStatistics` *calculate some statistics of the solutions of all haploblocks*

Description

calculate some statistics of the solutions of all haploblocks

Usage

```
calcStatistics(pedchk, ovwFS, indiv, haploblocks)
```

Arguments

`pedchk`	a list as generated by pedigreeHapCheck
`ovwFS`	a list as produced by overviewFS, or NA if no FS families available. If ovwFS is specified, pedchk and ovwFS should have been calculated from the same underlying data
`indiv`	missing, NULL or a vector of names of individuals. If present, the component pedstats is calculated only over the individuals in the vector, else over all individuals. Component FSstats is not affected by indiv
`haploblocks`	missing, NULL or a vector of names or numbers of haploblocks, or a list of haploblocks as in the inferHaplotypes input. If present, components pedstats and FSstats are calculated only over the specified haploblocks, else over all haploblocks.

Value

a list with one or two items:
pedstats: a data.frame with one row per haploblock, showing numbers of individuals in columns
- mrk: individuals with complete marker data
- hap: individuals with an assigned haplotype combination
- match.NA: individuals where a match between the haplotypes of the individual and its parents could not be verified (either the individual itself or both parents have no haplotype combination assigned)
- noDR.TRUE: individuals whose haplotype combination matches that of its parents, assuming polysomic inheritance without Double Reduction
- noDR.FALSE: individuals whose haplotype combination doesn't match that of its parents, assuming polysomic inheritance without Double Reduction
- withDR.TRUE: individuals whose haplotype combination matches that of its parents, assuming polysomic inheritance, allowing Double Reduction
- withDR.FALSE: individuals whose haplotype combination doesn't match that of its parents, assuming polysomic inheritance, allowing Double Reduction

The total number of individuals is
match.NA + noDR.TRUE + noDR.FALSE == match.NA + withDR.TRUE + withDR.FALSE;
the number of individuals with missing marker data or without assigned haplotype dosages can be calculated by subtracting mrk or hap from that total.

FSstats (only if ovwFS specified): a data.frame with for each FS family, "rest" and "all" one row, with columns
- pop: the FS family numbers, "rest" (all individuals not belonging to the FSs or their parents) and "all" (all individuals)
- bothparmrk: for how many haploblocks have both parents complete marker data?
- FSdone: for how many haploblocks has a solution based on polysomic inheritance been found for this FS family?
- mrk: the average number of (FS) individuals with complete marker data
- imp: the average number of (FS) individuals with imputed marker data
- hap: the average number of (FS) individuals with an assigned haplotype combination
The numbers for "all" include also the FS parents (each parent counted only once, even if some parents produced more than one FS family), which are not included in any of the other rows; therefore "all" is not equal to the sum of the rows above.

Examples

```
data(PolyHaplotyper_small)
phchk <- pedigreeHapCheck(ped=phped, mrkDosage=phdos, haploblock=phblocks,
                          hapresults=phresults)
phovw <- overviewByFS(haploblock=phblocks, parents=phpar, FS=phFS,
                      hapresults=phresults)
calcStatistics(pedchk=phchk, ovwFS=phovw)
```

---

checkmrkDosage *check a marker dosages matrix or data.frame*

Description

check a marker dosages matrix or data.frame, select columns and rows, convert to matrix

Usage

```
checkmrkDosage(mrkDosage, ploidy, indiv=NULL, markers=NULL,
generateMarkernames=TRUE)
```

Arguments

mrkDosage	matrix or data.frame. Markers are in rows, individuals in columns, each cell has a marker dosage. If mrkDosage is a matrix, the colnames are the individual names and the rownames are the marker names. If mrkDosage is a data.frame and the name of the first column starts with "marker" (upper/lowercase not relevant) the contents of this column are used as marker names, else the rownames are used as marker names. The (other) column names are the individual names. All marker dosages must be in 0:ploidy or NA.
ploidy	an integer value; all dosages are checked to be in 0:ploidy or NA
indiv	NULL (default) or a character vector with names of individuals to be selected. If NULL, all individuals are selected

markers NULL (default) or a character vector with names of markers to be selected. If NULL, all markers are selected.

generateMarkernames
if TRUE (default) and mrkDosage has no markernames specified, markernames are generated automatically. Markernames must be either present or be generated, and none may be missing.

Details

This function is called by inferHaplotypes and by mergeReplicates, so normally there is no need for the user to call this function directly.

Value

a matrix with the selected columns in the order of indiv and the selected rows in order of markers (or all columns or rows in the original order if indiv or markers are not specified), with names of individuals as column names, marker names as row names

Examples

```
mrkdos <- data.frame(
 marker=paste0("mrk", 1:3),
 indiv1=c(1, 2, 2),
 indiv2=c(4, 0, 0),
 indiv3=c(3, 0, 2))
# use all rows and columns:
checkmrkDosage(mrkDosage=mrkdos, ploidy=4)
# use only first and last row and column and change the order:
checkmrkDosage(mrkDosage=mrkdos, ploidy=4, indiv=c("indiv3", "indiv1"),
              markers=c("mrk3", "mrk1"))
```

---

compareHapMrkDosages *compare haplotyping results with observed markers dosages*

Description

compare haplotyping results with observed markers dosages

Usage

```
compareHapMrkDosages(mrkDosage, hapresults)
```

Arguments

mrkDosage a data.frame or matrix with the marker dosages, in the format of inferHaplotypes()

hapresults a list as returned by inferHaplotypes, with one item (itself a list) per haploblock with at least a matrix hapdos and a character vector markers. All markers in all haploblocks must also occur in mrkDosages, and all individuals in the hapdos matrices must also occur in mrkDosages

Value

a 3-D array with dimensions haploblock, individual, and chkresult: mrkNA (ALL markers in the haploblock have missing data for the individual), hapNA (the haplotype dosages do not sum to ploidy and/or are missing), match (TRUE if the non-missing marker dosages match the haplotype dosages, FALSE is there is a conflict, NA if mrkNA and/or hapNA are TRUE) Each element is itself a list with elements:

Examples

```
data(PolyHaplotyper_small)
chmd <- compareHapMrkDosages(mrkDosage=phdos, hapresults=phresults)
# show results for first haploblock, first 8 individuals:
chmd[1, 1:8,]
```

---

compareHapresults *compare two haplotyping results*

Description

compare two haplotyping results, e.g. PolyHaplotyper and SATlotyper

Usage

```
compareHapresults(haploblock, hapresultsA, hapresultsB)
```

Arguments

haploblock	a list of character vectors. The names are the names of the haploblocks, the character vectors have the names of the markers in each haploblock.
hapresultsA	and
hapresultsB	two list as returned by inferHaplotypes, with one item (itself a list) per haploblock with at least a matrix hapdos and a character vector markers. All haploblocks in param haploblock must occur in hapresultsA and in hapresultsB. The individual names (colnames of the hapdos items for each haploblock) must be identical and in the same order in hapresultsA and hapresultsB

Value

a list with one element per haploblock in param haploblock. Each element is itself a list with elements: $identical: TRUE or FALSE
$message: a single string, "" if the comparison is possible, else the reason why not (if $message is not "", $identical is always FALSE). The next elements are only present if $message is "":
$compindiv: a matrix comparing the two hapdos, with one column per individual and 5 rows: Both_NA, A_NA, B_NA, Equal, Uneq. The last 2 have NA values if A and/or B is NA $haplofreq: a matrix with one row per haplotype occurring in A and/or B, and columns A and B, with the total frequency of each haplotype in hapdos A or hapdos B

demo_ped *pedigree*

Description

A data.frame with a pedigree; used in vignette, stored in PolyHaplotyper_demo.RData

Format

a data.frame with 661 rows (samples) and 4 columns: genotype, mother, father, sample_nr

demo_snpdos *dosages of SNP alleles*

Description

A data.frame with the allele dosages for 30 SNPs in 661 samples; used in vignette, stored in PolyHaplotyper_demo.RData

Format

a data.frame with 30 rows (SNPs) and 663 columns. The first two columns contain the SNP name and contig; the remaining columns contain the allele dosages (integers in 0..6 or NA) of the SNPs in 661 samples

expandHapdos *Add dropped rows back to haplotype dosage matrix*

Description

Haplotype dosage matrices generated with the dropUnused=TRUE lack some haplotypes that are required by pedigreeHapCheck; this function adds them back.

Usage

```
expandHapdos(hapdos, nhap, hbname)
```

Arguments

hapdos	a haplotype dosage matrix as returned by a.o. inferHaplotypes. The rownames are assumed to have all the same length: they are composed of the haploblock name to which the 0-padded haplotype number is appended, separated by "_"
nhap	the total number of haplotypes for the haploblock: nhap = 2^nmrk with nmrk the number of markers in the haploblock
hbname	the haploblock name (only used when hapdos has 0 rows)

Value

a matrix similar to hapdos with rows for the dropped haplotypes re-inserted, with the correct rownames and containing only 0 and NA values

Examples

```
# specify haplotype dosages of 4 tetraploid individuals,
# only the 3 occurring haplotypes (1, 5 and 6) are given:
haplodosg <- matrix(c(1,2,1, 4,0,0, 0,4,0, 0,0,4), nrow=3,
                    dimnames=list(paste0("demohap_", c(1,5,6)), paste0("indiv", 1:4)))
# add the rows for the absent haplotypes, assuming the haploblock consists
# of 3 markers, so 8 haplotypes are possible:
expandHapdos(hapdos=haplodosg, nhap=8)
```

---

getFSfreqs *get all FS haplotype combinations expected from two parental haplotype combinations, with their frequencies*

Description

get all FS haplotype combinations expected from two parental haplotype combinations, with their frequencies. Different from getFScombs in that it returns the unique FS combinations as well as their frequencies, and that it also considers Double Reduction

Usage

```
getFSfreqs(parhac, DRrate)
```

Arguments

parhac	matrix with one column for each parent and <ploidy> rows, giving the haplotype combinations for each parent
DRrate	the rate of double reduction per meiosis (NOT per allele!); e.g. with a DRrate of 0.04, a tetraploid parent with genotype ABCD will produce a fraction of 0.04 of DR gametes AA, BB, CC and DD (each with a frequency of 0.01), and a fraction of 0.96 of the non-DR gametes AB, AC, AD, BC, BD, CD (each with a frequency of 0.16)

Value

a list of 2 elements:
FShac: a matrix with one column per unique FS haplotype combination and the same row count as parhac, giving the FS haplotype combinations. There are no duplicated columns but several (not necessarily adjacent) columns may correspond to the same mrkdid
freq: a vector of length ncol(hapcomb), with for each FS haplotype combination in hapcomb its frequency (0.0 ... 1.0). The colnames are NULL: haplotype combinations are not named.

Examples

```
# specify combinations of haplotypes in two tetraploid parents:
hapcomb <- matrix(c(2,2,5,6, 1,2,5,5), ncol=2)
# FS frequencies without double reduction:
getFSfreqs(parhac=hapcomb, DRrate=0)
# FS frequencies with 5\% double reduction:
getFSfreqs(parhac=hapcomb, DRrate=0.05)
```

`getGameteFreqs` *get all gametes and their frequencies for a parental haplotype combination*

Description

get all gametes and their frequencies for a parental haplotype combination

Usage

```
getGameteFreqs(parhac, DRrate)
```

Arguments

`parhac`	vector of the parental haplotype combination of length <ploidy>, giving the <ploidy> haplotype numbers present per haplotype giving the dosage of that haplotype, summing to ploidy
`DRrate`	the rate of double reduction per meiosis (NOT per allele!); e.g. with a DRrate of 0.04, a tetraploid parent with genotype ABCD will produce a fraction of 0.04 of DR gametes AA, BB, CC and DD (each with a frequency of 0.01), and a fraction of 0.96 of the non-DR gametes AB, AC, AD, BC, BD, CD (each with a frequency of 0.16)

Details

for hexaploids the DR gametes consist of a duplication of one of the 6 parental alleles, combined with one copy of one of the other 5 alleles.
Calculation is faster if DRrate is 0.0

Value

a list of 2 elements:
hapcomb: a matrix with one column per unique gamete and ploidy/2 rows. Each element is the number (ID) of a haplotype). Within columns the haplotypes are sorted from low to high; the columns are ordered from left to right, first on row 1, then on row 2 etc
freq: a vector of length ncol(hapcomb), with for each gamete in hapcomb its frequency (0.0 ... 1.0)

Examples

```
# specify combination of haplotypes in a tetraploid parent:
hapcomb <- c(2,2,5,6) # 2 copies of haplotype 2, 1 each of 5 and 6
# gamete frequencies without double reduction:
getGameteFreqs(parhac=hapcomb, DRrate=0)
# gamete frequencies with 5\% double reduction:
getGameteFreqs(parhac=hapcomb, DRrate=0.05)
```

---

hapcomb2hapdos *convert haplotype combinations to haplotype dosages*

Description

converts matrices that contain haplotype combinations in columns (as the matrices in ahccompletelist or ahclist) to matrices with the haplotype dosages

Usage

```
hapcomb2hapdos(hapcomb, nhap)
```

Arguments

hapcomb	matrix with <ploidy> rows and any number of columns. Each column contains a set of haplotypes (numbers in 1 ... nhap). NAs allowed but no values outside this range (not checked). E.g. a column with numbers 1-1-3-12 means two copies of haplotype 1 and one of haplotypes 3 and 12 each.
nhap	the total number of possible haplotypes: 2^nmrk where nmrk is the number of bi-allelic markers in the haploblock

Value

matrix with nhap rows (one row for each possible haplotype) and as many columns as in hapmat, giving the dosages of each haplotype in each column of hapmat.

Examples

```
# specify haplotype combinations of 2 individuals:
haplocomb <- matrix(c(1,5,5,6, 5,6,6,6), ncol=2,
                    dimnames=list(NULL, c("indiv1", "indiv2")))
# convert to dosage matrix,
# assuming haploblock has 3 markers, so 8 possible haplotypes:
hapcomb2hapdos(hapcomb=haplocomb, nhap=8)
```

hapdos2hapcomb *convert haplotype dosages to haplotype combinations*

Description

convert a vector or matrix of haplotype dosages to a vector or matrix of haplotype combinations

Usage

```
hapdos2hapcomb(hapdos, ploidy)
```

Arguments

hapdos	a vector with one item for some or all possible haplotypes, with the dosage of the haplotypes, summing to ploidy; or a matrix where each column is such a vector, each representing an individual. Missing data are allowed (but any NA makes the entire vector or matrix column unknown); if a matrix, 0 rows indicate missing values for all individuals. When hapdos is a matrix, the haplotype numbers must be given as rownames; when a vector, as names. Not all possible haplotype numbers need to be present but the ones that are present must be in ascending order, no duplicates or missing values allowed.
ploidy	the ploidy (one number)

Value

a sorted vector of length ploidy with all haplotype numbers present, or a matrix where each column is such a vector, with the same colnames as hapdos

Examples

```
# specify haplotype dosages of 4 tetraploid individuals,
# only the 3 occurring haplotypes (1, 5 and 6) are given:
haplodosg <- matrix(c(1,2,1, 4,0,0, 0,4,0, 0,0,4), nrow=3,
                    dimnames=list(paste0("demohap_", c(1,5,6)), paste0("indiv", 1:4)))
# usage with hapdos as matrix:
hapdos2hapcomb(hapdos=haplodosg, ploidy=4)
# usage with hapdos as vector:
hapdos2hapcomb(hapdos=haplodosg[, 1], ploidy=4)
```

hapdos2mrkdos *calculate the marker dosages resulting from haplotype dosage combinations*

Description

calculate the marker dosages resulting from haplotype dosage combinations

Usage

```
hapdos2mrkdos(hapdos, allhap)
```

Arguments

hapdos	a matrix with one column per combination of haplotypes and one row for each possible haplotype (corresponding to the rows of allhap) with dosage of the haplotypes in each combination. A vector is interpreted as a one-column matrix; all columns must sum to ploidy. The rownames of the matrix (or names of the vector) must contain the haplotype numbers
allhap	a matrix as returned by allHaplotypes

Details

if hapdos contains NA values, all values in the corresponding column of the result will also be NA

Value

a matrix with columns corresponding to the columns of hapdos and one row for each marker, with the dosages of each marker in each combination; colnames are the mrkdids (marker dosage IDs), rownames are the marker names taken from allhap

Examples

```
# get a matrix of all haplotypes with the 3 specified markers:
ah <- allHaplotypes(mrknames=c("mrkA", "mrkB", "mrkC"))
# specify haplotype dosages of 4 tetraploid individuals,
# only the 3 occurring haplotypes (1, 5 and 6) are given:
haplodosg <-
 matrix(c(1,2,1, 4,0,0, 0,4,0, 0,0,4), nrow=3,
        dimnames=list(paste0("demohap_", c(1,5,6)), paste0("indiv", 1:4)))
# calculate the corresponding marker (SNP) dosages:
hapdos2mrkdos(hapdos=haplodosg, allhap=ah)
```

haploblock_df2list *convert a haploblock-defining data frame to a list*

Description

convert a haploblock-defining data frame to a list as needed by inferHaplotypes

Usage

```
haploblock_df2list(df, mrkcol, hbcol, sorted=TRUE)
```

Arguments

df	a data.frame with at least the following two columns
mrkcol	the name or number of the column with the markers
hbcol	the name or number of the column with the haploblocks
sorted	if TRUE (default) the haploblock list is sorted in alphabetical order of haploblock name (the markers within haploblocks are not sorted); if FALSE the haploblocks will be in order of first occurrence in df

Details

function inferHaplotypes needs a list where each item is a vector of all markers in one haploblock. This function produces such a list from a data.frame where the markers are in one column and the haploblocks in another column. The markers and haploblocks columns may be character, factor or numeric, and the columns may be indicated by name or number.

Value

the desired list

Examples

```
df1 <- data.frame(
 marker=paste0("mrk",1:9),
 block=LETTERS[c(1,2,3,2,3,1,1,2,2)],
 extracol=runif(9))
haploblock_df2list(df=df1, mrkcol="marker", hbcol=2)
```

`inferHaplotypes` *infer haplotypes for one or more haploblocks*

Description

infer haplotypes for one or more haploblocks, for all individuals, using FS family(s) (with parents) if present, and infer haplotypes for non-FS material as well

Usage

```
inferHaplotypes(mrkDosage, indiv=NULL, ploidy, haploblock,
parents=NULL, FS=NULL, minfrac=c(0.1, 0.01), errfrac=0.025, DRrate=0.025,
maxmrk=0, dropUnused=TRUE, maxparcombs=150000, minPseg=1e-8,
knownHap=integer(0), progress=TRUE, printtimes=FALSE, ahcdir)
```

Arguments

`mrkDosage`	matrix or data.frame. Markers are in rows, individuals in columns, each cell has a marker dosage. Names of individuals are the column names, marker names are the row names or (if a data.frame) in a column named MarkerNames. All marker dosages must be in 0:ploidy or NA.
`indiv`	NULL (default) or a character vector with names of all individuals to be considered. If NULL, all columns of mrkDosage are selected. All indivs that are not in parents or FS vectors (see below) are considered unrelated, i.e. we have no implementation for pedigrees (yet).
`ploidy`	all marker dosages should be in 0:ploidy or NA
`haploblock`	a list of character vectors. The names are the names of the haploblocks, the character vectors have the names of the markers in each haploblock. Haplotype names are constructed from the haploblock names, which are used as prefixes to which the (zero-padded) haplotype numbers are are appended with separator '_'.
`parents`	a matrix with one row for each FS family and two columns for the two parents, containing the names of the female and male parent of each family.
`FS`	a list of character vectors. Each character vector has the names of the individuals of one FS family. The items of the list should correspond to the rows of the parents matrix, in the same order.
`minfrac`	vector of two fractions, default 0.1 and 0.01. A haplotype is considered to be certainly present if it must occur in at least a fraction minfrac[1] of all individuals; in the final stage for the "other" individuals (those that do not belong to the FS or its parents) this fraction is lowered to minfrac[2]; see also inferHaps_noFS
`errfrac`	default 0.025. The assumed fraction marker genotypes with an error (over all markers in the haploblock). The errors are assumed to be uniformly distributed over all except the original marker dosage combinations (mrkdids)

DRrate	default 0.025. The rate of double reduction per meiosis (NOT per allele!); e.g. with a DRrate of 0.04, a tetraploid parent with genotype ABCD will produce a fraction of 0.04 of DR gametes AA, BB, CC and DD (each with a frequency of 0.01), and a fraction of 0.96 of the non-DR gametes AB, AC, AD, BC, BD, CD (each with a frequency of 0.16)
maxmrk	Haploblocks with more than maxmrk markers will be skipped. Default 0: no haploblocks are skipped
dropUnused	TRUE (default) if the returned matrix should only contain rows for haplotypes that are present; if FALSE matrix contains rows for all possible haplotypes
maxparcombs	Parent 1 and 2 both may have multiple possible haplotype combinations. For each pair of haplotype combinations (one from P1 and one from P2) the expected FS segregation must be checked against the observed. This may take a long time if many such combinations need to be checked. This parameter sets a limit to the number of allowed combinations per haploblock; default 150000 takes about 45 min.
minPseg	default 1e-8. The minimum P-value of a chisquared test for segregation in FS families. The best solution for an FS family is selected based on a combination of P-value and number of required haplotypes, among all candidate solutions with a P-value of at least minPseg. If no such solution is found the FS and its parents are treated as unrelated material
knownHap	integer vector with haplotype numbers (haplotypes that must be present according to prior inference or knowledge, numbers refer to rows of matrix produced by allHaplotypes); default integer(0), i.e. no known haplotypes
progress	if TRUE, and new haplotype combinations need to be calculated, and the number of markers and the ploidy are both >= 6, progress is indicated by printed messages
printtimes	if TRUE, the time needed to process each haploblock is printed
ahcdir	a single directory, or not specified. inferHaplotypes uses lists that for each combination of marker dosages give all possible combinations of haplotype dosages. These lists (ahclist and ahccompletelist) are loaded and saved at the directory specified by ahcdir. If no ahcdir is specified it is set to the current working directory. If an ahclist or ahccompletelist for the correct ploidy is already in GlobalEnv this is used and no new list is loaded, even if ahcdir is specified.

Details

First we consider the case where one or more FS families and their parents are present in the set of samples. In that case, initially the possible haplotype configurations of the parents are determined. From that, all their possible gametes (assuming polysomic inheritance) are calculated and all possible FS haplotype configurations. Comparing this with the observed FS marker dosages the most likely parental and FS configurations are found.
It is possible that multiple parental combinations can explain the observed marker dosages in the FS. In that case, if one is clearly more likely and/or needs less haplotypes, that one is chosen. If there is no clear best solution still the parents and FS individuals that have the same haplotype configuration over all likely solutions are assigned that configuration.

For FS where no good solution is found (because of an error in the marker dosages of a parent, or because the correct solution was not considered) the parents and individuals will be considered as unrelated material.
If several FS families share common parents they are treated as a group, and only solutions are considered that are acceptable for all families in the group.
Finally (or if no FS families are present, immediately) the other samples are haplotyped, which are considered as unrelated material. If FS families have been solved the haplotypes in their parents are considered "known", and known haplotypes can also be supplied (parameter knownHap). For these samples we consecutively add haplotypes that must be present in a minimum number of individuals, always trying to minimize the number of needed haplotypes.
InferHaplotypes uses tables that, for each combination of dosages of the markers in the haploblock, list all haplotype combinations (ahc) that result in these marker dosages. In principle inferHaplotypes uses a list (ahccompletelist) that, for a given ploidy, has all the haplotype combinations for haploblocks from 1 up to some maximum number of markers. This list can be computed with function build_ahccompletelist. If this list is not available (or is some haploblocks contain more markers than the list), the ahc for the (extra) marker.
See the PolyHaplotyper vignette for an illustrated explanation.

Value

a list with for each haploblock one item that itself is a list with items:
message; if this is "" the haploblock is processed and further elements are present; else this message says why the haploblock was skipped (currently only if it contains too many markers)
hapdos: a matrix with the dosages of each haplotype (in rows) for each individual (in columns). For each individual the haplotype dosages sum to the ploidy. If dropUnused is TRUE Only the haplotypes that occur in the population are shown, else all haplotypes
mrkdids: a vector of the mrkdid (marker dosage ID) for each individual (each combination of marker dosages has its own ID; if any of the markers has an NA dosage the corresponding mrkdid is also NA).
The mrkdids can be converted to the marker dosages with function mrkdid2mrkdos.
markers: a vector with the names of the markers in the haploblock
imputedGeno: a matrix in the same format as param mrkDosage, with one row for each marker in the haploblock and one column per imputed individual, with the dosages of the markers. These are the individuals that have incomplete data in mrkDosage but where the available marker dosages match only one of the expected marker genotypes in the FS family (only individuals in FS families are imputed). It is possible that an individual with imputed marker dosages is not haplotyped (as is the case for individuals with complete marker data) if the marker dosages match different possible haplotype combinations. The next elements are only present if one or more FS families were specified:
FSfit: a logical vector with one element per FS family; TRUE if a (or more than one) acceptable solution for the FS is found (although if multiple solution are found they might not be used if unclear which one is the best solution). (Even if no solution was found for an FS, still its individuals may have a haplotype combination assigned ignoring their pedigree)
FSmessages: a character vector with one item per FS family: any message relating to the fitting of a model for that FS, not necessarily an error
FSpval: a vector of the chi-squared P-value associated with the selected FS model for each FS family, or the maximum P value over all models in case none was selected
If for new combinations of marker dosages the possible haplotype combinations have to be calculated, an ahclist file is written to ahcdir

Examples

```
# this example takes about 1 minute to run:
data(PolyHaplotyper_small)
results <- inferHaplotypes(mrkDosage=phdos, ploidy=6,
haploblock=phblocks, parents=phpar, FS=phFS)
names(results)
names(results[[1]])
```

`make.Happyinf.input` *convert PolyHaplotyper input data to Happy-inf format*

Description

convert PolyHaplotyper input data to Happy-inf format for a single haploblock

Usage

```
make.Happyinf.input(mrkDosage, indiv=NULL, haploblock,
ploidy, fname)
```

Arguments

`mrkDosage`	matrix or data.frame of allele dosages; same as input for inferHaplotypes. Markers are in rows, individuals in columns, each cell has a marker dosage. All marker dosages must be in 0:ploidy or NA
`indiv`	the names of the individuals to include in the Happy-inf input data. Default NULL includes all individuals
`haploblock`	a list of character vectors. The names are the names of the haploblocks, the character vectors have the names of the markers in each haploblock. Only the markers in haploblock will be included in the Happy-inf input data
`ploidy`	single integer: the ploidy level
`fname`	filename of a tab-separated output file: this will contain the data in Happy-inf format (the saved data.frame is also the return value). If "" no file is written

Value

a data.frame in the Happy-inf input format: a header row with "SNPID", "block" and names of the individuals and one row per marker, with only the individuals, markers and blocks as specified. #'SNPID has the marker names, "Block" the haploblock names. All markers of a haploblock are in contiguous rows. Missing dosages are represented by "NA"

Examples

```
data(PolyHaplotyper_small)
HAPin <- make.Happyinf.input(mrkDosage=phdos, haploblock=phblocks,
                             ploidy=6, fname="")
HAPin[,1:8]
```

`make.SATlotyper.input` *convert PolyHaplotyper marker data to SATlotyper format*

Description

convert PolyHaplotyper marker data (input) to SATlotyper format for a single haploblock.

Usage

```
make.SATlotyper.input(mrkDosage, indiv=NULL, markers,
ploidy, phenotype=0, fname)
```

Arguments

`mrkDosage`	matrix or data.frame of allele dosages; same as input for inferHaplotypes. Markers are in rows, individuals in columns, each cell has a marker dosage. All marker dosages must be in 0:ploidy or NA
`indiv`	the names of the individuals to include in the ShesisPlus input data. Default NULL includes all individuals
`markers`	character vector with the names of the markers in the haploblock; all must occur in mrkDosage
`ploidy`	single integer: the ploidy level
`phenotype`	vector with the phenotypes of all individuals, in order of the columns of dosmat; default 0
`fname`	filename of a *.csv output file: this will contain the data in the SATlotyper format (the saved data.frame is also the return value). If "" no file is written

Value

a data.frame in the SATlotyper input format: a header row with "Genotype" and the marker names, and one row per individual with the individual name plus for each marker the genotype as a sorted string of <ploidy> A's and B's, or <ploidy> N's

Examples

```
data(PolyHaplotyper_small)
SATin <- make.SATlotyper.input(mrkDosage=phdos, markers=phblocks[[1]],
                               ploidy=6, fname="")
head(SATin)
```

`make.ShesisPlus.input` *convert PolyHaplotyper marker data to ShesisPlus format*

Description

convert PolyHaplotyper marker data (input) to ShesisPlus format for a single haploblock.

Usage

```
make.ShesisPlus.input(mrkDosage, indiv=NULL, markers,
ploidy, phenotype=0, fname="")
```

Arguments

`mrkDosage`	matrix or data.frame of allele dosages; same as input for inferHaplotypes. Markers are in rows, individuals in columns, each cell has a marker dosage. All marker dosages must be in 0:ploidy or NA
`indiv`	the names of the individuals to include in the ShesisPlus input data. Default NULL includes all individuals
`markers`	character vector with the names of the markers in the haploblock; all must occur in mrkDosage
`ploidy`	single integer: the ploidy level
`phenotype`	vector with the phenotypes of all individuals, in order of the columns of dosmat; default 0
`fname`	filename of the output file: this will contain the data in the ShesisPlus format (the saved data.frame is also the return value). If "" (default) no file is written

Details

ShesisPlus needs the data formatted as: 1 column with names of individuals, 1 column with the phenotypes (with missing values represented as "NA"), and for each of the selected markers <ploidy> columns, with the (unphased) marker alleles. Here we use only biallelic markers; their alleles are indicated by 1 and 2 for the ref and alt allele, and 0 for missing data.
The contents of file fname can be pasted into the input data box of the ShesisPlus web interface at http://shesisplus.bio-x.cn/SHEsis.html

Value

a data.frame in the described ShesisPlus input format

Examples

```
data(PolyHaplotyper_small)
SSPin <- make.ShesisPlus.input(mrkDosage=phdos, markers=phblocks[[1]],
                               ploidy=6)
SSPin[1:6,1:8]
```

mergeReplicates *merge replicate samples in dosage matrix*

Description

merge replicate samples in dosage matrix

Usage

```
mergeReplicates(mrkDosage, replist, solveConflicts=TRUE)
```

Arguments

mrkDosage	a dosage matrix with markers in rows and individuals in columns. row names are marker names, column names are individual names.
replist	a list of character vectors, each of which has the sample names of a set of replicates
solveConflicts	if TRUE (default) and there are conflicting dosage assignments between replicates for the same marker, the one with highest frequency is used, provided the total freq of other dosages = 1 OR <= 10% of the frequency of the most frequent dosage. If solveConflicts is FALSE and there are conflicting dosages, the consensus for that marker will be NA

Details

This function merges all sets of replicates, each to one column. The column name of the one retained column is the first one for that set in replist.
For each set of replicates it calls getConsensusmrkDosage

Value

a version of mrkDosage in which only one column of each set of replicates is retained; this column (the first in its set as specified in replist) now has the consensus scores over all replicates. Also, if mrkDosage was a data.frame, it is converted into a matrix.

Examples

```
# construct a dosage matrix with some missing data:
dosmat <-
 matrix(c(rep(c(3,0,1),3), rep(c(1,1,2),4)), nrow=3,
        dimnames=list(c("mrk1","mrk2","mrk3"),
                      c("a1","a2","a3", "b1","b2","b3","b4")))
ix <- matrix(c(1,1, 3,1, 2,2, 1,3, 2,4, 1,5, 2,6), ncol=2, byrow=TRUE)
dosmat[ix] <- NA
dosmat
# define 2 sets of replicates:
reps <- list(c("a1","a2","a3"), c("b1","b2","b3","b4"))
# merge:
```

```
mergeReplicates(mrkDosage=dosmat, replist=reps)
# introduce a conflicting dosage:
dosmat[3,2] <- 2
# merge:
mergeReplicates(mrkDosage=dosmat, replist=reps)
```

`mrkdid2mrkdos` *get the marker dosages from mrkdids (marker dosage IDs)*

Description

get the marker dosages from mrkdids (marker dosage IDs)

Usage

```
mrkdid2mrkdos(dosageIDs, nmrk, ploidy, mrknames=NULL)
```

Arguments

`dosageIDs`	vector of marker-dosage-combination IDs (mrkdid)
`nmrk`	character vector of marker names in the haploblock
`ploidy`	the ploidy level, a single positive integer
`mrknames`	a vector of nmrk marker names (default NULL): if not NULL these are used as rownames of the returned matrix

Value

a matrix with in columns the marker dosages corresponding to the marker dosageIDs, with these mrkdids as colnames, and one row per marker, with marker names as rownames if mrknames are specified

Examples

```
# dosages of 3 markers in 3 tetraploid individuals:
mrkdosg <-
 matrix(c(1,2,2, 4,0,0, 3,0,2), nrow=3,
        dimnames=list(c("mrkA", "mrkB", "mrkC"), c("indiv1", "indiv2", "indiv3")))
# get the "marker dosage IDs":
dids <- mrkdos2mrkdid(mrkDosage=mrkdosg, ploidy=4)
# convert dids back to marker dosages:
mrkdid2mrkdos(dosageIDs=dids, nmrk=3, ploidy=4, mrknames=c("mrkA", "mrkB", "mrkC"))
```

mrkdos2mrkdid *get marker dosage IDs from marker dosages*

Description

get marker dosage IDs (mrkdid) from marker dosages

Usage

```
mrkdos2mrkdid(mrkDosage, indiv=NULL, ploidy, check=TRUE)
```

Arguments

mrkDosage	matrix or data.frame. Markers are in rows, individuals in columns, each cell has a marker dosage. Names of individuals are the column names, marker names are the row names or (if a data.frame) in a column named MarkerNames. All marker dosages must be in 0:ploidy or NA. If a data.frame, additional columns may be present.
indiv	NULL (default) or a character vector with names of individuals to be selected. If NULL, all columns are selected; if mrkDosage is a data.frame, that is probably not what is intended.
ploidy	all marker dosages are checked to be in 0:ploidy or NA
check	if TRUE (default) checkmrkDosage is called. If FALSE it is assumed that mrkDosage is a matrix (not a data.frame) and it is not checked.

Details

with ploidy==1 and (of course) all dosages 0 or 1 this function returns the haplotype numbers for the haplotype specified by each column

Value

a vector of marker dosage IDs, one for each column of mrkDosage: each a number in 1:((ploidy+1)^nrow(mrkDosage)), NA for each column in dosages where any of the dosages are NA

Examples

```
# dosages of 3 markers in 3 tetraploid individuals:
mrkdosg <-
 matrix(c(1,2,2, 4,0,0, 3,0,2), nrow=3,
        dimnames=list(c("mrkA", "mrkB", "mrkC"), c("indiv1", "indiv2", "indiv3")))
# get the "marker dosage IDs":
dids <- mrkdos2mrkdid(mrkDosage=mrkdosg, ploidy=4)
# convert dids back to marker dosages:
mrkdid2mrkdos(dosageIDs=dids, nmrk=3, ploidy=4, mrknames=c("mrkA", "mrkB", "mrkC"))
```

`overviewByFS` *generate an overview of the results by haploblock and by FS family*

Description

generate an overview of the results by haploblock and by FS family

Usage

```
overviewByFS(haploblock, parents, FS, hapresults)
```

Arguments

`haploblock`	a list of character vectors. The names are the names of the haploblocks, the character vectors have the names of the markers in each haploblock.
`parents`	a matrix of two colums and one row per FS, containing the names of the two parents of each FS
`FS`	a list of character vectors with the names of the samples for each FS
`hapresults`	a list as returned by inferHaplotypes, with one item per haploblock, containing at least all those in the list specified by haploblock

Value

a list with two items:
- ovw: an integer matrix with one row per haploblock and the following columns:
* nmrk: the number of markers in the haploblock
* nhap: the number of different haplotypes assigned over all individuals (NA if no solution was found for this haplotype; the reason for that is listed in the first column of item messages of the return value)
* for each FS family a set of 6 columns:
+ parmrk (0, 1 or 2: the number of parents with complete marker data)
+ fit (0=FALSE or 1=TRUE), indicating if a solution for the FS was found based on polysomic inheritance)
+ mrk: the number of FS progeny with complete marker data
+ imp: the number of FS progeny where complete marker data were imputed
+ hap: the number of FS progeny with assigned haplotype combinations. hap will be less than the mrk value if the same FS marker genotype can be produced with different combinations of haplotypes that are all compatible with the parental haplotype combinations) or if some FS marker genotypes cannot be produced by the inferred parental haplotype combinations
+ P: the chi-squared P-value of the best fitting solution, even if this is discarded because of lack of fit).
* For "rest" (all individuals that are not part of the FS's or their parents) and "all" (all individuals) there are also columns mrk and hap, and for "all" there is also a column imp, similar to those for the FS families. The numbers for "all" are the sums of those for the FS families, the FS parents (some FS may share a parent but shared parents are counted only one) and the "rest".
- messages : a character matrix with one row per haploblock and the following columns:

* haploblock: the reason why there is no solution for the haploblock ("" if there is a solution) * one column for each FS family with a possible message or "". A message can indicate a failure to find a solution for the FS family but may also describe less significant problems, such as some progeny with unexpected marker dosages etc.

Examples

```
data(PolyHaplotyper_small)
overviewByFS(haploblock=phblocks, parents=phpar, FS=phFS,
            hapresults=phresults)
```

---

padded *pad an integer (prefix with zeroes to a fixed length)*

Description

pad an integer (prefix with zeroes to a fixed length)

Usage

```
padded(x, maxx=0)
```

Arguments

x	vector of non-negative integers
maxx	a single integer to whose nchar all x elements will be padded; if 0 (default) the largest value in x will be used

Value

a character vector representing the values of x left-padded with 0's to the length of integer maxx or of max(x)

Examples

```
padded(c(21, 1, 121, NA, 0))
padded(c(21, 1, 121, NA, 0), maxx=1000)
```

pedigreeHapCheck *check the consistency of haploblock genotypes over a pedigree*

Description

For all haploblocks, check whether the inheritance of inferred haplotypes over the pedigree is consistent without or with allowing for double reduction (DR), and assuming polysomic inheritance.

Usage

```
pedigreeHapCheck(ped, mrkDosage, haploblock, hapresults)
```

Arguments

ped	data.frame or matrix listing the pedigree. Column 1 has names of all individuals (no duplicates, no NA), column 2 and 3 have the parents (duplicates and NA allowed, also individuals with one known parent or with two identical parents allowed). No sorting of the pedigree is needed. All parents should also be listed as individuals; if that is not the case the missing parents will be added as founders and a warning will be issued.
mrkDosage	a matrix with one row for each marker and one column for each individual, with the observed dosages. Individuals with one or more NA dosages are considered to have no marker data. Individual names are the colnames, no duplicates allowed; the set of individuals needs not be the same as those in ped; individual names common to mrkDosage and ped must be exactly identical (upper/lower case and whitespace included)
haploblock	a list of character vectors. The names are the names of the haploblocks, the character vectors have the names of the markers in each haploblock
hapresults	a list as returned by inferHaplotypes, with one item per haploblock, containing at least all those in the list specified by haploblock

Value

a list with two items:
- ped_arr: a 3D logical array with dimensions individuals, diagnostics and haploblocks. For each individual and each haploblock there are 4 diagnostics:
* mrk: does the individual have complete marker dosage data?
* imp: were the marker dosages for this individual imputed?
* hap: is there a haplotype combination assigned to the individual?
* noDR: does the haplotype genotype of the individual match that of its parents, assuming polysomic inheritance but no Double Reduction? NA if the individual or both its parents do not have a haplotype genotype assigned
* withDR: as noDR, but allowing Double Reduction
- parents_arr: a 3D integer array with dimensions parents (all individuals that occur as parents in the pedigree), diagnostics and haploblocks. For each parent and each haploblock there are 7 diagnostics:

* par_mrkdata: 0=FALSE, 1=TRUE, does this parent have complete marker data?
* par_hapdata: 0=FALSE, 1=TRUE, does this parent have a haplotype genotype assigned?
* totprogeny: how many first-generation progeny (children) does this parent have (combined over all its matings, both as mother and as father)
* prog_mrkdata: how many progeny have complete marker dosage data?
* prog_hapdata: how many progeny have a haplotype combination assigned?
* nonDRmatch: how many progeny have a haplotype combination that is compatible with their parent's haplotype combinations, assuming polysomic inheritance but no Double Reduction
* DRmatch: as nonDRmatch, but also allowing DR

Both ped_arr and parents_arr contain all haploblocks in haploblock, also those skipped because of too many markers and those without any haplotyped individuals. These can be excluded by excluding them from the haploblock list.

Examples

```
data(PolyHaplotyper_small)
phchk <- pedigreeHapCheck(ped=phped, mrkDosage=phdos, haploblock=phblocks,
                          hapresults=phresults)
# show the top of the ped_arr for haploblock 1:
phchk$ped_arr[1:6,,1]
# show the top of the parents_arr for haploblock 1:
phchk$parents_arr[1:6,,1]
```

---

pedigreeSim2PH *convert the PedigreeSim true haplotypes to marker dosages and haplotype dosages*

Description

convert the PedigreeSim true haplotypes to marker dosages and haplotype dosages

Usage

```
pedigreeSim2PH(ps_geno, haploblock, indiv=NULL, dropUnused=TRUE)
```

Arguments

ps_geno	the filename of a *_genotypes.dat file as produced by PedigreeSim, or a data.frame read from such a file (with a column marker, followed by <ploidy> columns per individual with the marker alleles for each homolog; the names of these columns must be <indivname>_<homolog number>). ps_geno should not contain missing data (this will be true unless the PedigreeSim output is midified)
haploblock	a list of character vectors. The names are the names of the haploblocks, the character vectors have the names of the markers in each haploblock. Haplotype names are constructed from the haploblock names, which are used as prefixes to which the (zero-padded) haplotype numbers are are appended with separator '_'. All markers in haploblock must be present in ps_geno. If haploblock is NULL only the marker dosages are generated, not the haplotype dosages.

indiv	the names of the individuals to be extracted; default NULL: all individuals in ps_geno. If not NULL, all indiv must be present in ps_geno
dropUnused	TRUE (default) if the returned matrix should only contain rows for haplotypes that are present; if FALSE matrix contains rows for all possible haplotypes

Details

if all alleles are in 0/1, these are kept. If only 2 allele symbols occur in ps_geno all are converted (alphabetically) to 0/1 in the same way (e.g. if the alleles are A and B, A -> 0 and B -> 1, even in markers with only B's). If different allele symbols are used between markers, per marker the (alphabetically or numerically) lowest -> 0 and the highest -> 1. So if more than two different allele symbols occur in ps_geno, and one marker has only A and another only B alleles, both are converted to 0's.
in mrkDosage, the dosage of the alleles (converted to) 1 is reported

Value

a list with 2 items:
$mrkDosage: a matrix of marker dosages in the input format of inferHaplotypes, with all markers that occur in haploblock, sorted according to haploblock, and for the selected indiv in the specified order
$haplist: (only if haploblock is not NULL) a list with one element for each haploblock, similar to the inferHaplotypes output). Each element is itself a list, with two components:
$hapdos is a matrix with the haplotype dosages for that haploblock for each individual
$markers: a vector with the names of the markers in the haploblock in the output of inferHaplotypes

---

phblocks *List of markers per haploblock*

Description

A list with for each haploblock the names of the markers it contains; used in manuals, stored in PolyHaplotyper_small.RData

Format

a list with 2 character vectors

phdos *dosages of SNP alleles*

Description

A matrix with the allele dosages for 8 SNPs in 661 samples; used in manuals, stored in PolyHaplotyper_small.RData

Format

a data.frame with 30 rows (SNPs) and 663 columns. The first two columns contain the SNP name and contig; the remaining columns contain the allele dosages (integers in 0..6 or NA) of the SNPs in 661 samples

phFS *members of FS families*

Description

A list with for each FS family the names of the individuals it contains; used in manuals, stored in PolyHaplotyper_small.RData

Format

a list with 4 vectors, each with the (here numeric) names of the FS members

phpar *parents of FS families*

Description

A matrix with the names of the parents of each FS family; used in manuals, stored in PolyHaplotyper_small.RData

Format

a matrix with 4 rows, 1 per FS family, and 2 columns, 1 for each parent

phped *pedigree*

Description

A data.frame with a pedigree; used in manuals, stored in PolyHaplotyper_small.RData

Format

a data.frame with 661 rows (samples) and 4 columns: genotype, mother, father, sample_nr

phresults *haplotyping results*

Description

A list with the results of function inferHaplotypes; used in manuals, stored in PolyHaplotyper_small.RData

Format

a list with 2 elements, one per haploblock; each element itself a list with multiple components

read.Happyinf.output *read the haplotyping results from the Happy-inf output*

Description

read the haplotyping results from the Happy-inf output

Usage

```
read.Happyinf.output(file_prefix, dropUnused=TRUE)
```

Arguments

file_prefix	the prefix for the output files generated by Happy-inf (the value of the -o parameter)
dropUnused	TRUE (default) if the returned matrix should only contain rows for haplotypes that are present; if FALSE matrix contains rows for all possible haplotypes

Details

This function reads the <file_prefix>.stat.dat and <file_prefix>.stats.dat files. The first contains one row per SNP marker, grouped per haploblock and <ploidy> columns per individual with the SNP haplotypes, where SNP alleles are represented as 0 or 1, or NA for unknown. For one SNP, the <ploidy> alleles in an individual are all known or all unknown It is possible that of the different SNPs in a haploblock some are known and some are not; in the conversion to PolyHaplotyper format all these partially known haplotypes are made unknown. The latter file has a statistics "mismatch" and "ratio" for each individual / haploblock combination; for their meaning see the Happyinf readme file.

Value

a list with 2 items:
$hapdos is a matrix with individuals in columns and haplotypes in rows, giving the dosages of the haplotypes in each individual (summing to ploidy). This is the same format as the hapdos components of the inferHaplotypes results of PolyHaplotyper except that
$stats is a matrix with individuals in rows (matching dosmat) and two rows named mismatch and ratio.

---

read.SATlotyper.output

read the haplotyping results from the SATlotyper output

Description

read the haplotyping results from the SATlotyper output

Usage

```
read.SATlotyper.output(fname, output="", allelecodes=c("A", "B"),
sep, haploblockname="")
```

Arguments

fname	filename of an xml file produced by SATlotyper
output	character vector, console output of SATlotyper; this contains a table of calculated haplotypes with more info than the xml file
allelecodes	the codes used for the ref and alt SNP alleles in make.SATlotyper.input, default "A" and "B"
sep	the separator used in make.SATlotyper.input (SATlotyper cleverly identifies this separator and uses it for its output), default tab
haploblockname	if not "" (default) the haplotype IDs are given as <haploblockname>_<hapnr>, else just as <hapnr>, left_padded with zeroes

Details

The xml file is parsed using the package XML. The resulting list has 3 items named source, bootstrapping and haplotyping. In this function source is ignored and the results are obtained from haplotypings[[1]] (i.e. even if multiple haplotypings were done only the first is extracted), with the haplotype score from the bootstrapping item added to the haplotype.info.

Value

a list with 3 items:
$hapdos is a matrix with individuals in columns and haplotypes in rows, giving the dosages of the haplotypes in each individual (summing to ploidy). This is the same format as the hapdos components of the inferHaplotypes results of PolyHaplotyper
$haplotype.info is a data.frame that is a combination of information from two or three tables:
columns Haplotype and necessity are from element haplotypings[[1]]$haplotypes in the xml file,
column hapnr gives the PolyHaplotyper equivalents of the Haplotypes,
column score is from element bootstrapping in the xml file,
columns id, number, frequency, homozygous, necessary, distance and neighbours are from (the console) output if present.
The haplotype.info data.frame is ordered by hapnr.
$HaplotypingScore is the single number in $haplotyping[[1]]$score.

`read.ShesisPlus.output` *Read the haplotyping results from the ShesisPlus output*

Description

Read the haplotyping results from the ShesisPlus output

Usage

```
read.ShesisPlus.output(SSPout, order.by=c("", "hapnr", "count")[2])
```

Arguments

`SSPout`	filename of a text file with ShesisPlus output (as copied from the web page produced by running ShesisPlus web interface at http://shesisplus.bio-x.cn/SHEsis.html); or a character vector with the same output
`order.by`	how to order the rows of the hapstat data.frame. "hapnr" means ordering by haplotype number, "count" means ordering by decreasing Total.count, anything else results in no reordering. By default order by hapnr.

Details

If present, the markernames and the haplotype statistics are read from the file. ShesisPlus does not provide haplotype combinations for individuals

Value

a list with 2 items: $markernames has the marker names in the markers in the haploblock (if present in the file); hapstat contains the haplotype statistics as read from the file with an additional (first) column hapnr: the haplotype numbers as defined in PolyHaplotyper. The other columns are Haplotype (as sequences of marker alleles), Total.count (of the haplotype over the whole population) and, according to an email from Zhiqiang Li of 13-04-2020) BETA: Regression coefficient, SE: Standard error, R2: Regression r-squared, T: t-distribution statistics, P: p-value

Examples

```
# we give a typical SSP output as character vector; instead we could also
# give the name of a text file
SSPout <- c(
 " Please cite:",
 "",
 "    Shen, J. et al. SHEsisPlus, a toolset for genetic studies ...",
 "    Shi, Y. et al. SHEsis, a powerful software platform ...",
 "    Li, Z. et al. A partition-ligation-combination-subdivision ...",
 "",
 "if you find this tool useful in your research. Thanks!",
 "",
 "Haplotype Analysis:",
 "",
 "Haplotypes with frequency <0.03 are ignored.",
 "Loci chosen for haplotype analysis: m1, m2, m3, m4",
 "Haplotype  Total count  Beta  SE  R2  T  p",
 "1122  2232  0.002  0.01  1.01e-04  0.252  0.8",
 "1222  230  -0.019  0.017  0.002  -1.15  0.25",
 "1221  152  -0.01  0.024  2.94e-04  -0.43  0.667",
 "2222  288  0.008  0.02  2.93e-04  0.429  0.667",
 "1121  142  -0.009  0.022  2.81e-04  -0.42  0.674"
)
read.ShesisPlus.output(SSPout)
```

---

run.SATlotyper *A simple interface to run SATlotyper*

Description

A simple interface to run SATlotyper

Usage

```
run.SATlotyper(path_to_SATlotyper, infile, outfile,
SAT_solver="sat4j.conf")
```

Arguments

path_to_SATlotyper	path to the folder where SATlotyper.jar and the *.conf files of the SAT-solvers are located
infile	name (and path) to the SATlotyper input file
outfile	name (and path) for the SATlotyper output file (an xml file)
SAT_solver	name of the *.conf file for the SAT-solver to use. Default "sat4j.conf" because this works under both Windows and Linux.

Details

This function issues a system command to invoke SATlotyper. Java and SATlotyper must be installed. This is just a simple interface for convenience; for more control run SATlotyper directly from the command window.

Value

The return is a list with elements:
$cmd : the command passed to the system
$result: screen output from SATlotyper; this contains some extra info not present in the output xml file)
The main result is the outfile

---

showOneFS *show marker and haplotype dosages for one FS family*

Description

show marker and haplotype dosages for one FS family and its parents

Usage

```
showOneFS(FSnr, hbresults, mrkDosage, FS, parents)
```

Arguments

FSnr	the number of the FS family (indexes the FS list and parents)
hbresults	a list with the haplotyping results for one haploblock: one element of a list as returned by inferHaplotypes
mrkDosage	a matrix of marker dosages, may contain rows for more markers than only those in the current haploblock (the relevant markers are specified in the hbresults list)
FS	a list of which each item is a (character) vector with the names of the individuals in that FS family
parents	a (character) matrix with 2 columns and one row for each FS family in FS, with the names of the two parents of each family

Value

a list with 3 elements:
$mrkdat: a matrix with info on the marker dosages distribution in the FS. The first two columns are for the parents: the parent name is between (brackets) in the column name, their mrkdid (marker dosage ID) in row 1 and their marker dosages below that. The remaining columns are for the different mrkdids observed in the FS: the mrkdid itself in the column name, its frequency in row 1 and its marker dosages below that. The final column gives the frequency of individuals with one or more missing marker dosages.
$hapdat: a matrix with similar layout as mrkdat, but now with the haplotype dosages rather than the marker dosages. Some mrkdids (columns) may not have a haplotype dosage combination assigned (if multiple possible haplotype combinations result in the same marker dosages)
usedhap: a matrix with the dosage (0 or 1) of each marker in each of the used haplotypes; haplotype nrs in columns, markers in rows

Examples

```
data(PolyHaplotyper_small)
# show the results of the first FS family in the first haploblock:
showOneFS(FSnr=1, hbresults=phresults[[1]], mrkDosage=phdos,
         FS=phFS, parents=phpar)
```

---

totHapcombCount *calculate the total nr of possible haplotype combinations*

Description

calculate the total nr of possible haplotype combinations

Usage

```
totHapcombCount(ploidy, nmrk)
```

Arguments

ploidy	a vector of 1 or more ploidy levels
nmrk	a vector of 1 or more numbers of markers per haploblock

Details

nmrk is used to calculate the total number of possible haplotypes = 2^nmrk. The shorter vector of ploidy and nmrk is recycled.

Value

a vector with the number of possible haplotype combinations for each pair of ploidy and nmrk values

Examples

```
totHapcombCount(ploidy=4, nmrk=c(1:8))
```

---

usedhap *Find all used (inferred) haplotypes*

Description

Find all haplotypes for a haploblock that were inferred to be present in the population (i.e. all haplotypes used for haplotyping any of the individuals)

Usage

```
usedhap(hapresults, haploblock)
```

Arguments

hapresults	list as returned by inferhaplotypes, or one element of such a list (i.e. the results for one haploblock)
haploblock	if hapresults is one element of the return value of inferHaplotypes, haploblock should be missing of NULL; else haploblock is a single value indicating the haploblock: either its name of its index in hapresults

Details

This function works with the results of inferHaplotypes; the setting of dropUnused does not affect this function

Value

an array with the haplotypes that are used in the population. The haplotypes are in columns, with the haplotype numbers as colnames; the markers are in rows.

Examples

```
data(PolyHaplotyper_small)
# show the composition of haplotypes inferred to be present
# in the first haploblock:
usedhap(hapresults=phresults, haploblock=1)
```

Index

∗ **datasets**
  demo_ped, 10
  demo_snpdos, 10
  phblocks, 30
  phdos, 31
  phFS, 31
  phpar, 31
  phped, 32
  phresults, 32

allhap, 2
allHaplotypes, 3

build_ahccompletelist, 4

calcMrkHaptable, 5
calcStatistics, 6
checkmrkDosage, 7
compareHapMrkDosages, 8
compareHapresults, 9

demo_ped, 10
demo_snpdos, 10

expandHapdos, 10

getFSfreqs, 11
getGameteFreqs, 12

hapcomb2hapdos, 13
hapdos2hapcomb, 14
hapdos2mrkdos, 15
haploblock_df2list, 16

inferHaplotypes, 17

make.Happyinf.input, 20
make.SATlotyper.input, 21
make.ShesisPlus.input, 22
mergeReplicates, 23
mrkdid2mrkdos, 24
mrkdos2mrkdid, 25

overviewByFS, 26

padded, 27
pedigreeHapCheck, 28
pedigreeSim2PH, 29
phblocks, 30
phdos, 31
phFS, 31
phpar, 31
phped, 32
phresults, 32

read.Happyinf.output, 32
read.SATlotyper.output, 33
read.ShesisPlus.output, 34
run.SATlotyper, 35

showOneFS, 36

totHapcombCount, 37

usedhap, 38