Package 'IBDhaploRtools'

February 19, 2015

Type Package

Title Functions for the Analysis of IBD Haplo Output

Version 1.8

Date 2015-01-27

Author Marshall Brown, Fiona Grimson

Maintainer Fiona Grimson <fgrimson@uw.edu>

Description Functions to analyze, plot, and store the output of running IBD_Haplo software package. More information regarding IBD_Haplo can be found at http://www.stat.washington.edu/thompson/Genepi/pangaea.shtml.

License GPL-3

LazyLoad yes

LazyData yes

Repository CRAN

Repository/R-Forge/Project morgan-rtools

Repository/R-Forge/Revision 36

Repository/R-Forge/DateTimeStamp 2015-02-14 01:16:14

Date/Publication 2015-02-17 01:07:43

Depends R (>= 2.10)

NeedsCompilation no

R topics documented:

`IBDhaploRtools-package`

Functions for the Analysis of IBD Haplo output

Description

IBDhaploRtools consists of several functions to analyze, plot, and store the output of the IBD_Haplo software package. More information regarding IBD_Haplo can be found at www.stat.washington.edu/thompson/Genepi/pang

Details

Package:	IBDhaploRtools
Type:	Package
Version:	1.8
Date:	2015-01-27
License:	GPL-3
LazyLoad:	yes

See the package vignette: vignette("IBDhaploRtools_tutorial")

Author(s)

Marshall D. Brown

Fiona Grimson <fgrimson@uw.edu>

References

none

Examples

```
# See the tutorial that  outlines the use of all the
```

```
# functions found in this package:
# vignette( IBDhaploRtools_tutorial )
```

`cumul.sum` *cumulative.sums*

Description

Calculate cumulative sum

Usage

```
cumul.sum(x)
```

Arguments

`x`	x is a vector

Value

Returns a vector y where y[i]=sum(x[<i])

Examples

```
x = 1:10
cumul.sum(x)
```

`get.counts` *get counts*

Description

subroutine that counts transitions

Usage

```
get.counts(x, state.num)
```

Arguments

`x`	vector of data
`state.num`	scalar indicating number of states

Value

returns a state.num by state.num matrix of transition counts

Examples

```
x = sample( 1:5, size = 20, replace = TRUE)
get.counts(x, 5)
```

---

h.to.g *Convert haplotype states to genotype states*

Description

Convert haplotype states (1 - 15) to genotype states (1 - 9)

Usage

```
h.to.g(hap.states)
```

Arguments

hap.states	vector or matrix of haplotype states. Values should be integers from 1-9

Value

vector or matrix of same dimension as hap.states but with the corresponding genotypic states.

Author(s)

F L Grimson

Examples

```
## this example is taken from the package vignette.
##See vignette(IBDhaploRtools_tutorial)

data(trueibd_phased)
trueibd_unphased <- h.to.g( trueibd_phased )
```

ibdhap.barplot *create ibd barplot*

Description

Graphically displays regions of any ibd sharing, no ibd sharing and no calls along a chromosome for a set of haplotypes / pair of genotypes.

Usage

```
ibdhap.barplot(x, data.type = c("h", "g", "r"), col = c("grey", "red",
"white"), position = NA, top = 1, bottom = 0, density = 50, ...)
```

Arguments

x	A vector of ibd states (with values 0 - 15 for haplotypic, 0-9 genotypic, one for each marker). This is expected to be a single column taken from the output of ibdhap.make.states.
data.type	"h" : haplotypic data "g" : genotypic data (or hap data ran as genotypic) "r" : reduced data (output from running ibdhap.reduce.states and then ibdhap.make.states)
col	Vector consisting of three colors. The colors represent no calls, any ibd shared, and no ibd shared, respectively.
position	A position vector, with the same length as x describing the positions (in cM, M, or any other metric) of each marker. If positions is not included, "segment length" refers to number of SNPs making up a segment.
top	top value of rectangles in barplot
bottom	bottom value of rectangles in barplot
density	density of diagonal lines filling each segment in lines per sq inch.
...	additional graphical parameters

Author(s)

MD Brown

Examples

```
## this example is taken from the package vignette.
##See vignette(IBDhaploRtools_tutorial)

data(qibd_phased)
data(ids_phased)

phased.gold <- ibdhap.make.calls( qibd.file  = qibd_phased,
                  ids.file= ids_phased, cutoff = 0.8)
```

```
par(mfrow=c(4,1))
ibdhap.barplot(phased.gold[,1], data.type="h", xlab="", ylab="")
ibdhap.barplot(phased.gold[,2], data.type="h", xlab="", ylab="")
ibdhap.barplot(phased.gold[,3], data.type="h", xlab="", ylab="")
ibdhap.barplot(phased.gold[,4], data.type="h", xlab="", ylab="")
```

`ibdhap.compare.loci` *ibdhap compare.loci*

Description

Compares inferred ibd state with simulated "true" states. Calculates the proportion of markers called in the correct state, false positives (i.e. inferring ibd shared when none is shared), false negatives (i.e. inferring no ibd shared when ibd sharing is present) and the proportion of no calls.

Usage

```
ibdhap.compare.loci(calls, true, data.type)
```

Arguments

`calls`	The data.frame created from running ibdhap.make.calls on ibd_haplo output.
`true`	The data frame of same dimension as calls, but with true (probably simulated) ibd states.
`data.type`	"h" : haplotypic data "g" : genotypic data (or hap data ran as genotypic) "r" : reduced data (output from running ibdhap.reduce.states and then ibdhap.make.states)

Value

Returns a list consisting of three matrices called "all", "ibd", "nonibd" and "categories".

The matrix for "all" contain a row for each of the following quantities, for all the loci:

`Number of sites`

`Called Correctly`

`Called IBD Incorrectly`

`Called noIBD Incorrectly`

The matrix for "ibd" contains a row for each of the following quantities, based on just the loci that are truly in an IBD state:

`Number of sites`

`Called Correctly`

Called as wrong IBD

Called as no IBD

No-call

The matrix for "nonibd" contains a row for each of the follwing quantities, based on just the loci that are truly in an non-ibd state:

Number of sites

Called Correctly

Called as IBD

No-call

Finally, the matrix for "categories" tabulates the percentage of the loci that are truly in, and called to be in, each IBD state.

Author(s)

Fiona Grimson

Examples

```
## this example is taken from the package vignette.
##See vignette(IBDhaploRtools_tutorial)

data(qibd_phased)
data(ids_phased)
data(trueibd_phased)

phased.gold <- ibdhap.make.calls( qibd.file  = qibd_phased,
                  ids.file= ids_phased, cutoff = 0.8)

ibdhap.compare.loci(phased.gold, trueibd_phased, "h")
```

---

ibdhap.compare.segs *ibdhap compare segments*

Description

Compares inferred ibd states with simulated "true" states. Calculates segments of ibd in the true data, giving descriptions of the segments and the proportion of correct and incorrect calls within the segments.

Usage

```
ibdhap.compare.segs(calls, true, data.type, seg.cutoff, pos=NA)
```

Arguments

calls	The data.frame created from running ibdhap.make.calls on ibd_haplo output.
true	The data frame of same dimension as calls, but with true (probably simulated) ibd states.
data.type	"h" : haplotypic data "g" : genotypic data (or hap data ran as genotypic) "r" : reduced data (output from running ibdhap.reduce.states and then ibdhap.make.states)
seg.cutoff	A scalar value between 0 and 1 to act as the cutoff value, that is, the percentage of loci in the segment whose calls must agree to call the segment for a particular ibd state.
pos	A position vector with the same length as the number of loci (rows of calls or true) describing the positions in cM, M or any other metric of each marker. If positions are not included, the segment lengths reported will be the number of markers making up a segment

Value

Returns a list containing two elememts "seg.stats", "seg.info".

The "seg.stats" matrix has a row for each of the following statistics

`Number of segments`

`Number of IBD segments`

`IBD segs called correctly`

`IBD segs called no-IBD`

`IBD segs called wrong IBD`

`IBD segs with no call`

The "seg.info" matrix has a row for each segment of IBD in the true data and a column for each of the following statistics for each segment.

`seg.length`

`true.state`

`seg.call`

`mode.call`

`prop.corr`

Author(s)

Fiona Grimson

Examples

```
## this example is taken from the package vignette,
## See vignette(IBDhaploRtools_tutorial)

data(qibd_phased)
data(ids_phased)
data(trueibd_phased)

phased.gold <- ibdhap.make.calls( qibd.file  = qibd_phased,
                  ids.file= ids_phased, cutoff = 0.8)

ibdhap.compare.segs(phased.gold, trueibd_phased, "h", 0.8, pos=NA)
```

`ibdhap.make.calls` *ibdhap.make.calls*

Description

Stores and simplifies the qibd files created by IBD Haplo by "calling" a marker to be in an ibd state if it's marginal probability meets some "cutoff" value, a zero or "no call" is assigned to a marker in which no single state meets the value assigned to "cutoff". The R data.frame that this function creates is expected by other functions in this package.

Usage

```
ibdhap.make.calls(qibd.filename = NULL, ids.filename = NULL, qibd.file
= NULL, ids.file = NULL, cutoff = 0.8)
```

Arguments

`qibd.file`	A matrix of the contents of the qibd.out file the is produced from running ibd_haplo. If this is left as the default NULL, qibd.filename should be specified
`qibd.filename`	The filename (location) of the qibd.out file the is produced from running ibd_haplo. This is to be input as a character string. The qibd file will be loaded from this location if qibd.file is not specified.
`ids.file`	A matrix of the .ids file used to run ibd_haplo.If this is left as the default NULL, ids.filename should be specified
`ids.filename`	The filename (location) of the ids file used to run ibd_haplo. This is to be input as a character string. The ids file will be loaded from this location if ids.file is not specified.
`cutoff`	A scalar value between 0 and 1 to act as the "cutoff" value. This is the value which, if the maximum marginal probability of an ibd state is greater than, that marker will be called that state. Otherwise, the marker is called as zero, which in this context, means that there was not evidence enough to determine the specific ibd state of that marker. Default value is 0.8.

Value

Returns a data.frame with ncol = # of sets of haplotypes/ pairs of genotypes in the qibd file. nrow = # of markers/SNPs Each column of this data.frame consists of integers (0 - 15 for haplotypes, 0-9 for genotypic data) corresponding to the ibd state at that marker (if the probability of that state for the marker is maximal and exceeds "cutoff" value, or a 0 value (for no call).

Note

This data.frame is required for all other functions in this package, so calling this function first is required.

Author(s)

M.D. Brown

Examples

```
## this example is taken from the package vignette.
##See vignette(IBDhaploRtools_tutorial)

data(qibd_phased)
data(ids_phased)

phased.gold <- ibdhap.make.calls( qibd.file  = qibd_phased,
                  ids.file= ids_phased, cutoff = 0.8)

## alternatively, specify the file location, e.g.
## qibd.filename <- '~/Documents/qibd_unphased_2011.gold'
## ids.filename <- '~/Documents/ids_unphased_2011.gold'
## phased.gold <- ibdhap.make.calls( qibd.filename = qibd.filename,
##                   ids.filename = ids.filename, cutoff = 0.8)
```

---

ibdhap.make.true *ibdhap make true states*

Description

This function reads ibd_haplo output of true (simulated) ibd states into an R matrix data structure. The true states are compared to the state calls.

Usage

```
ibdhap.make.true( true.filename )
```

Arguments

true.filename	The filename of the true pairwise ibd states from, for example, outfifteenibd.txt or outnineibd.txt

Value

An R matrix data structure with a column for each pairwise comparison in the input file. The pair names are thrown away.

Examples

```
## For an existing file called "outfifteenibd.txt" use
## ibdhap.make.true( "outfifteenibd.txt" )

## An example of a data set already read into R is
## data( trueibd_phased )
```

`ibdhap.names` *Get chromosome names*

Description

Get identifying infomation of chromosomes making up each set, this consists of a number identifying the individual and 0/1 indicator of which of their two chromosomes was used.

Usage

```
ibdhap.names(ids.file=NULL, ids.filename = NULL)
```

Arguments

`ids.file`	A matrix of the .ids file used to run ibd_haplo.If this is left as the default NULL, ids.filename should be specified
`ids.filename`	The filename (location) of the ids file used to run ibd_haplo. This is to be input as a character string. The ids file will be loaded from this location if ids.file is not specified.

Value

Matrix with a row for each set of chromosomes and two columns for each constituent chromosome. The first column is the individual number and the second column indicates which of their two chromosomes is used.

Author(s)

F L Grimson

Examples

```
## this example is taken from the package vignette.
##See vignette(IBDhaploRtools_tutorial)

data(ids_phased)
ibdhap.names( ids.file = ids_phased )
```

`ibdhap.reduce.states` *ibdhap reduce states*

Description

This function reduces the columns of a qibd.out file created by ibd_haplo by summing probabilities over certain columns. When reducing an output file that was run on haplotypes, this script will sum over columns 1-8 (other), 9-10 (2 pairs chrs. ibd), 11-14 (one pair of chrs. ibd),and 15 (no ibd). When reducing a file from genotypic date, the corresponding columns are summed over so that they reflect the same values (other, one pair ibd, two pairs ibd, not ibd). A file will then be output with these probabilites displayed just like the original ibd_haplo output qibd file.

Usage

```
ibdhap.reduce.states(qibd.filename, dat.filename, output.filename)
```

Arguments

`qibd.filename`	The filename of the qibd.out file the is produced from running ibd_haplo. This is to be input as a character string. In the examples, this file is called "qibd_g.gold".
`dat.filename`	The filename of the .dat file used to run ibd_haplo. This is the complicated parameter file that consists of three lines of values / indicators that tells ibd_haplo what to do. In the examples, this files is called "compu_4haps.dat".
`output.filename`	The name of the file that this function will print to. In the examples, this filename is "test.file.txt".

Value

A txt document the looks exactly like the qibd file input, but the ibd state probabilities per marker are summed over in the manner described above.

Author(s)

MD Brown

Examples

```
## See vignette("IBDhaploRtools_tutorial" )
```

ibdhap.seg.lengths *ibdhap.seg.lengths*

Description

Given the ibd states from a set of haplotypes/pair of genotypes (taken from a column of the output of ibdhap.make.states), this function creates a data.frame consisting of all segments of differing ibd state, paired with their respective length.

Usage

```
ibdhap.seg.lengths(x, position = NA)
```

Arguments

x	A vector of ibd states (with values 0 - 15 for haplotypic, 0-9 genotypic, one for each marker). This is expected to be a single column taken from the output of ibdhap.make.states.
position	A position vector, with the same length as x describing the positions (in cM, M, or any other metric) of each marker. If positions is not included, "segment length" refers to number of SNPs making up a segment.

Value

A data.frame with 2 columns and nrow = nrow(x)(number of markers). column1 is the integer value corresponding to ibd state, column2 is the length of the segment for that state as measured by the positions vector.

Author(s)

MD Brown

Examples

```
## The function is currently defined as
function( x, position=NA ){
#x is a single column from ibdhap.make.states output
## NOT INCLUDING THE haplotype/genotype names!,
# thus it is the ibd.states from one pair of individuals (genotypes)
# or set of 4 haplotypes


n.marker<- length(x)  #number of markers

#if positions are given, we use them, otherwise "length" refers to
# number of SNPs
if(is.element(position,NA)){position <- 1:(n.marker) }
```

```
# obtain a vector of ibd state change points --index where ibd state changes
change.points<-c(1)

for(imarker in 2:n.marker)
{
prev.val<-x[imarker-1]
val <- x[imarker]

if( prev.val!=val)
{
   change.points=c(change.points, imarker)
}

}

# tidy up the end of change.points
  if(change.points[length(change.points)]!= n.marker){change.points=c(change.points, n.marker)}

change.points.pos<-position[change.points]
seg.lengths<-diff(change.points.pos)
ibd.state<-x[change.points[1:length(seg.lengths)] ]

  return( as.data.frame(cbind(ibd.state = ibd.state, seg.lengths = seg.lengths)))

}
```

`ibdhap.summary.calls` *ibdhap summary of called states*

Description

Summarizes the data created by ibdhap.make.states by giving mean lengths of ibd segments, mean proportions of ibd shared, and counts on ibd segments. This gives information on the group of sets of haplotypes/genotypes or on an individual pairing.

Usage

```
ibdhap.summary.calls(calls, data.type = c("h", "g", "r"), position = NA)
```

Arguments

`calls`	The data.frame created from running ibdhap.make.calls on ibd_haplo output.
`data.type`	"h" : haplotypic data "g" : genotypic data (or hap data ran as genotypic) "r" : reduced data (output from running ibdhap.reduce.states and then ibdhap.make.states)
`position`	A position vector, with the same length as nrow(states.dat) describing the positions (in cM, M, or any other metric) of each marker. If positions is not included, "segment length" refers to number of SNPs making up a segment.

Details

ibdhap.summary analyzes all the data it is given.If states.dat consists of more than one column, means are calculated across all sets of haplotypes. For summaries on only one set of haplotypes/pair of genotypes, just pass this function the column of data corresponding to the specific set on which you want summary statistics.

Value

Returns a list consisting of:

`mean.prop`	mean proportion of chromosomes called in any ibd shared, no ibd shared, and no calls
`mean.length`	mean lengths of segments of chromosome called in any ibd shared, no ibd shared, and no calls. Length is measured by the position vector (see above).
`seg.counts`	total counts of segments of chromosome called in any ibd shared, no ibd shared, and no calls

Author(s)

MD Brown

Examples

```
## this example is taken from the package vignette.
##See vignette(IBDhaploRtools_tutorial)

data(qibd_phased)
data(ids_phased)
data(trueibd_phased)

phased.gold <- ibdhap.make.calls( qibd.file  = qibd_phased,
                   ids.file= ids_phased, cutoff = 0.8)

summary.phased <- ibdhap.summary.calls( phased.gold, data.type="h")
```

`ibdhap.transitions` *create transition matrix*

Description

Creates a matrix of transition counts from when ibd state switches along the chromosome.

Usage

```
ibdhap.transitions(calls, data.type = c("h", "g", "r"))
```

Arguments

calls	The data.frame created from running ibdhap.make.calls on ibd_haplo output.
data.type	"h" : haplotypic data "g" : genotypic data (or hap data ran as genotypic) "r" : reduced data (output from running ibdhap.reduce.states and then ibdhap.make.states)

Details

To create this matrix, all no calls are ignored. This is because, when transitioning out of being relatively certain of an ibd state, the marginal probabilities of ibd state by marker usually move into a segment of uncertainty (hence no calls) before it becomes relatively certain of an ibd state and therefore switches states.

Value

A matrix of size 15 x 15 (haplotypic) or 9 x 9 (genotypic) that shows counts of ibd state transitions. Element [i,j] of the output is the number of times state i changed to state j in the data.

Author(s)

MD Brown

Examples

```
## this example is taken from the package vignette.
##See vignette(IBDhaploRtools_tutorial)

data(qibd_phased)
data(ids_phased)
data(trueibd_phased)

phased.gold <- ibdhap.make.calls( qibd.file  = qibd_phased,
                  ids.file= ids_phased, cutoff = 0.8)
transitions.phased <- ibdhap.transitions(phased.gold, data.type="h")
```

---

ids_phased *Example MORGAN IBDhaplo output*

Description

These files are example output from the MORGAN IBDhaplo program. There is an ids and qibd file produced in each IBDhaplo run. This output is from two runs, each on the same original data one of which was genotypic (unphased) and one of which was haplotypic (phased). The true ibd states of the data are given in trueibd_phased, in terms of haplotypic IBD states.

There is also the vector posvec which gives the positions of each of the 2000 SNPs used in the data files.

Please refer to the MORGAN IBDhaplo documentation for information on how to generate these files and what they contain.

Usage

```
ids_phased
```

Format

ids files contain a 4x11 matrix, qibd files contain an 8000x18 matrix.

Source

MORGAN IBDhaplo

References

http://www.stat.washington.edu/thompson/Genepi/pangaea.shtml

`make.col.vec` *make color vector*

Description

utility function to make a vector of colors used for plotting

Usage

```
make.col.vec(x, colors)
```

Arguments

`x`	vector of 0, 1, 2's
`colors`	vector of "col1", "col2" , "col3"

Value

returns vector of colors

`meet.cutoff` *meet cutoff*

Description

determines if a number is larger than a cutoff

Usage

```
meet.cutoff(num, cutoff)
```

Arguments

`num`	number to be tested
`cutoff`	cutoff to be tested against

Value

TRUE/FALSE

`removezeros` *remove zeros*

Description

remove zeros from a vector

Usage

```
removezeros(X)
```

Arguments

`X`	a vector

Value

the same vector but with zeros removed

Examples

```
X = c(1,0,2,0,3)

removezeros(X)

## equivalently
 X[X!=0]
```

return.ibd.val *return ibd value*

Description

utility function that returns the index of a 1, if there is one in this vector

Usage

```
return.ibd.val(col.dat)
```

Arguments

col.dat	a vector of all zeros with at most a single 1

sumcol *sum over specific columns*

Description

utility function to sum over pre-defined entries of a vector

Usage

```
sumcol(rowdat)
```

Arguments

rowdat	a vector of length 15

Value

a vector of length 4, since some elements were summed over

Index