

Package ‘autohaplo’

March 24, 2017

Type Package

Title What the Package Does (Title Case)

Version 0.1

Date 2017-03-14

Author Who wrote it

Imports snpStats, VariantAnnotation, reshape2, ggplot2, LDcorSV

Maintainer Who to complain to <yourfault@somewhere.net>

Description More about what it does (maybe more than one line)

License What license is it under?

LazyData TRUE

RoxygenNote 6.0.1

R topics documented:

across.center.ld	2
calc.ld	2
combine.snpObject	3
distance.filter	3
genotype.filter	4
get.max.ld.across.center	4
get.min.interval	5
ld.block	5
ld.block.each.side	6
marker.selection	6
marker.subset	7
one.side.ld	7
parse.hapmap	8
plot.chromosome	8
plot.distance.distribution	9
plot.ld.distance	9
plot.ld.matrix	10
position.filter	10
position.subset	11
read.vcf	11
snpObject.output	12
split.snpObject	12
write.hapmap	13
write.vcf	13

Index**14**

across.center.ld	<i>Drop markers which are not in LD with at least one other marker across the center position.</i>
------------------	--

Description

Drop markers which are not in LD with at least one other marker across the center position.

Usage

```
across.center.ld(snpObject, center.pos, ld.threshold = 0.5,
  max.pair.distance = 1e+10, max.marker.distance = 1e+10)
```

Arguments

snpObject	The snpObject to be filtered.
center.pos	The central position across which genes must be in LD.
ld.threshold	The minimum LD value to pass the filter. (default: 0.5)

Value

A subset of the original SnpObject containing only markers in LD with another marker across the center position.

calc.ld	<i>Calculate linkage disequilibrium and keep the resulting matrix in the SnpObject.</i>
---------	---

Description

Calculate linkage disequilibrium and keep the resulting matrix in the SnpObject.

Usage

```
calc.ld(snpObject, measure = "r2")
```

Arguments

snpObject	The snpObject whose LD matrix must be computed.
-----------	---

Value

The original SnpObject, with an additional item containing the LD matrix.

combine.snpObject	<i>Combines two snpObjects into one.</i>
-------------------	--

Description

Combines two snpObjects into one.

Usage

```
combine.snpObject(snpObject.5, snpObject.3, original.snpObject = NULL)
```

Arguments

snpObject.5	The snpObject of markers on the 5' side of the gene.
snpObject.3	The snpObject of markers on the 3' side of the gene.

Value

A single snpObject combining both sets of markers.

distance.filter	<i>Drop markers which are past a certain relative distance from the central position.</i>
-----------------	---

Description

Drop markers which are past a certain relative distance from the central position.

Usage

```
distance.filter(snpObject, center.pos, ld.threshold = 0.5,  
  iqr.threshold.multiplier = 1.5)
```

Arguments

snpObject	The snpObject to be filtered.
center.pos	The central position on either side of which markers must be in LD.
ld.threshold	The minimum LD value to pass the filter. (default: 0.5)
iqr.threshold.multiplier	The multiplier to be applied to the IQR when determining the maximum distance from the central position. (default: 1.5)

Value

A subset of the original SnpObject containing only markers whose distance from other markers are within a certain multiplier of the IQR of all distances between markers.

genotype.filter	<i>Filters the markers from a SnpObject based on their overall genotypes.</i>
-----------------	---

Description

Genotypes are filtered according to three criteria:

Chromosome Only markers on the chromosome given by chrom are kept.

Homozygosity Markers with heterozygous phenotypes are removed.

Frequency Only markers where the alternative allele has a frequency higher than min.freq.threshold are kept.

Usage

```
genotype.filter(snpObject, chrom,
  min.alt.threshold = (1/nrow(snpObject$Matrix)), max.het.threshold = 0,
  max.missing.threshold = 0)
```

Arguments

snpObject	The snpObject to be filtered.
chrom	The chromosome for which markers should be kept.
min.freq.threshold	The minimum frequency of the alternative allele necessary for keeping the marker. Defaults to two individuals.

Value

A subset of the original SnpObject containing only those markers which passed the filters.

get.max.ld.across.center	<i>Determine the maximum LD between markers across the gene center.</i>
--------------------------	---

Description

Determine the maximum LD between markers across the gene center.

Usage

```
get.max.ld.across.center(snpObject, center.pos)
```

Arguments

snpObject	The snpObject whose max LD must be evaluated.
center.pos	The central position on either side of which markers must be in LD.

Value

The maximum LD value between any two markers across the gene center.

get.min.interval	<i>Subset a snpObject based upon the IQR of markers and the max R^2.</i>
------------------	---

Description

Filters genes markers by distance, only keeping those that are within $((2 - \text{max.ld}) * \text{IQR})$ of the gene center.

Usage

```
get.min.interval(snpObject, center.pos)
```

Arguments

snpObject	The snpObject whose max LD must be evaluated.
center.pos	The central position on either side of which markers must be in LD.

Value

A subset of the original snpObject whose markers fall within the specified range.

ld.block	<i>Drop markers in perfect LD with adjacent markers, keeping only a single representative marker.</i>
----------	---

Description

Drop markers in perfect LD with adjacent markers, keeping only a single representative marker.

Usage

```
ld.block(snpObject, block.threshold = 1, reverse = FALSE, verbose = FALSE)
```

Arguments

snpObject	The snpObject to be filtered.
block.threshold	The minimum R^2 for two adjacent markers to be eligible for blocking (default: 1).
reverse	If true, blocks start at the most 3' position.

Value

A subset of the original SnpObject containing only one representative marker per LD block.

ld.block.each.side	<i>Perform LD clustering on either side of the gene.</i>
--------------------	--

Description

Clustering is performed separately on 5' and 3' markers, each starting from the central position. Results are then combined and returned.

Usage

```
ld.block.each.side(snpObject, center.pos, block.threshold = 1)
```

Arguments

snpObject	The snpObject to be filtered.
center.pos	The central position on either side of which clustering should be performed.
block.threshold	The minimum R^2 for two adjacent markers to be eligible for blocking (default: 1).

Value

A subset of the original SnpObject containing only one representative marker per LD block.

marker.selection	<i>NA</i>
------------------	-----------

Description

Filters gene markers by distance, only keeping those that are within $((2 - \text{max.ld}) * \text{IQR})$ of the gene center.

Usage

```
marker.selection(analysis.parameters)
```

Arguments

input.file	The name of the input file containing the genotype calls.
gene.chrom	The chromosome of the gene of interest.
gene.start	The start position of the gene of interest.
gene.end	The end position of the gene of interest.
chrom.length	The length of the chromosome on which the gene of interest lies.
label	A label for generating output files.
file.format	A string specifying the file format of the output. Can be "auto", "vcf" or "hapmap".

Value

A snpObject containing the subset of the chosen markers.

marker.subset	<i>Subsets a snpObject by keeping only certain markers.</i>
---------------	---

Description

Subsets a snpObject by keeping only certain markers.

Usage

```
marker.subset(snpObject, indices)
```

Arguments

snpObject	The snpObject to be subset.
indices	Indices of the markers to be kept.

Value

A subset of the original SnpObject containing only those markers in indices.

one.side.ld	<i>Drop markers which are not in LD with at least one other marker on their side of the gene divide.</i>
-------------	--

Description

Drop markers which are not in LD with at least one other marker on their side of the gene divide.

Usage

```
one.side.ld(snpObject, center.pos, ld.threshold = 0.5, max.distance = 1e+10)
```

Arguments

snpObject	The snpObject to be filtered.
center.pos	The central position on either side of which markers must be in LD.
ld.threshold	The minimum LD value to pass the filter. (default: 0.5)
max.distance	The maximum distance between a pair of markers for them to be kept. Defaults to 1e10, which should be more than any known chromosome.

Value

A subset of the original SnpObject containing only markers in LD with one other marker on their side of the center position.

parse.hapmap	<i>Parse a hapmap file.</i>
--------------	-----------------------------

Description

Parse a hapmap file.

Usage

```
parse.hapmap(file.name)
```

Arguments

file.name	The file to be parsed
-----------	-----------------------

Value

A list containing two elements:

Matrix A matrix of class [SnpMatrix](#) with the observed genotypes.

Metadata A data frame with metadata about the imported markers.

plot.chromosome	<i>Generates a plot showing the position of markers along the chromosome.</i>
-----------------	---

Description

Generates a plot showing the position of markers along the chromosome.

Usage

```
plot.chromosome(snpObject, center.pos, chr.length,
  filename = "Markers on chromosome.pdf", zoom = FALSE)
```

Arguments

snpObject	The snpObject to be plotted.
center.pos	The central position on either side of which markers must be in LD.
chr.length	The length of the chromosome.
filename	Filename for the output plot.

`plot.distance.distribution`

Plots a distribution of the distances between markers on either side of the gene.

Description

Plots a distribution of the distances between markers on either side of the gene.

Usage

```
plot.distance.distribution(snpObject, center.pos,  
  filename = "Distance distribution.pdf")
```

Arguments

snpObject	The snpObject to be exported.
filename	Filename for the output file.

`plot.ld.distance`

Generates a plot showing the relationship between LD and distance between markers.

Description

Generates a plot showing the relationship between LD and distance between markers.

Usage

```
plot.ld.distance(snpObject, center.pos, filename = "LD vs distance.pdf")
```

Arguments

snpObject	The snpObject to be plotted.
center.pos	The central position on either side of which markers must be in LD.
filename	Filename for the output plot.

plot.ld.matrix	<i>Generates a tile plot of the LD of the given snpObject.</i>
----------------	--

Description

Generates a tile plot of the LD of the given snpObject.

Usage

```
plot.ld.matrix(snpObject, center.pos, filename = "LD matrix.pdf")
```

Arguments

snpObject	The snpObject to be plotted.
center.pos	The central position on either side of which markers must be in LD.
filename	Filename for the output plot.

position.filter	<i>Drop markers based on the overall distribution of marker positions.</i>
-----------------	--

Description

This function establishes the quartiles of the position distribution, and eliminates markers that are outside of a certain number of inter-quartile ranges.

Usage

```
position.filter(snpObject, iqr.threshold.multiplier = 2)
```

Arguments

snpObject	The snpObject to be filtered.
iqr.threshold.multiplier	The multiplier to be applied to the IQR when determining the maximum and minimum positions. (default: 2)

Value

A subset of the original SnpObject containing only markers within the desired interval.

position.subset	<i>Subsets a snpObject by keeping only markers within a certain interval.</i>
-----------------	---

Description

Subsets a snpObject by keeping only markers within a certain interval.

Usage

```
position.subset(snpObject, position.5, position.3)
```

Arguments

snpObject	The snpObject to be subset.
position.5	The upstream extremity of the interval.
position.3	The downstream extremity of the interval.

Value

A subset of the original SnpObject containing only those markers between position.5 and position.3.

read.vcf	<i>Parse a vcf file.</i>
----------	--------------------------

Description

Parse a vcf file.

Usage

```
read.vcf(file.name)
```

Arguments

file.name	The file to be parsed
-----------	-----------------------

Value

A list containing two elements:

Matrix A matrix of class [SnpMatrix](#) with the observed genotypes .

Metadata A data frame with metadata about the imported markers.

snpObject.output	<i>Output both the hapmap/vcf export and the graphs.</i>
------------------	--

Description

Output both the hapmap/vcf export and the graphs.

Usage

```
snpObject.output(snpObject, center.pos, chr.length, label, file.format,
  output.markers = FALSE, output.matrix = FALSE, output.chrom = FALSE,
  output.chrom.zoom = FALSE, output.distance.between.markers = FALSE,
  output.ld.vs.distance = FALSE, output.markers.density = FALSE,
  verbose = FALSE)
```

Arguments

snpObject	The snpObject to be plotted.
center.pos	The central position on either side of which markers must be in LD.
chr.length	The length of the chromosome.
label	A label from which to generate filenames.
file.format	The file format to use for output. Can be vcf of hapmap.

split.snpObject	<i>Splits an snpObject in its 5' and 3' components relative to the gene.</i>
-----------------	--

Description

Splits an snpObject in its 5' and 3' components relative to the gene.

Usage

```
## S3 method for class 'snpObject'
split(snpObject, center.pos)
```

Arguments

snpObject	The snpObject to be split.
center.pos	The central position on either side of which the matrix must be split.

Value

A list with two snpObject, "FivePrime" and "ThreePrime", containing the markers on either side of the central position.

write.hapmap	<i>Export an SnpObject into the hapmap format.</i>
--------------	--

Description

Export an SnpObject into the hapmap format.

Usage

```
write.hapmap(snpObject, filename)
```

Arguments

snpObject	The snpObject to be exported.
filename	Filename for the output file.

write.vcf	<i>Write VCF output</i>
-----------	-------------------------

Description

This will generate a VCF file containing the genotypes. This will only work if the initial file was in VCF format.

Usage

```
write.vcf(snpObject, filename)
```

Arguments

snpObject	The object to be written to the file.
filename	The name of the output file.

Index

`across.center.ld`, [2](#)

`calc.ld`, [2](#)

`combine.snpObject`, [3](#)

`distance.filter`, [3](#)

`genotype.filter`, [4](#)

`get.max.ld.across.center`, [4](#)

`get.min.interval`, [5](#)

`ld.block`, [5](#)

`ld.block.each.side`, [6](#)

`marker.selection`, [6](#)

`marker.subset`, [7](#)

`one.side.ld`, [7](#)

`parse.hapmap`, [8](#)

`plot.chromosome`, [8](#)

`plot.distance.distribution`, [9](#)

`plot.ld.distance`, [9](#)

`plot.ld.matrix`, [10](#)

`position.filter`, [10](#)

`position.subset`, [11](#)

`read.vcf`, [11](#)

`SnpMatrix`, [8](#), [11](#)

`snpObject.output`, [12](#)

`split.snpObject`, [12](#)

`write.hapmap`, [13](#)

`write.vcf`, [13](#)