Package 'LDheatmap'

October 12, 2022

Version 1.0-6

Date 2022-05-03

Title Graphical Display of Pairwise Linkage Disequilibria Between SNPs

Maintainer Brad McNeney <mcneney@sfu.ca>

Depends R (>= 4.0)

Imports grid, genetics, snpStats, Rcpp

Suggests rtracklayer, GenomicRanges, GenomeInfoDb, IRanges, lattice,
ggplot2, vcfR, RColorBrewer

Description Produces a graphical display, as a heat map, of measures
of pairwise linkage disequilibria between single nucleotide polymorphisms (SNPs). Users may
optionally include the physical locations or genetic map
distances of each SNP on the plot. The methods are described in
Shin et al. (2006) <doi:10.18637/jss.v016.c03>. Users should note that the
imported package 'snpStats' and the suggested packages
'rtracklayer', 'GenomicRanges', 'GenomInfoDb' and 'IRanges' are all
BioConductor packages <https://bioconductor.org>.

License GPL-3

URL https://sfustatgen.github.io/LDheatmap/

Repository CRAN

Date/Publication 2022-04-24 00:20:02 UTC

RoxygenNote 7.1.2

LinkingTo Rcpp

NeedsCompilation yes

Author Ji-Hyung Shin [aut],
Sigal Blay [aut],
Nicholas Lewin-Koh [ctb],
Brad McNeney [aut, cre],
Gloria Yang [ctb],
Matthew Reyers [ctb],
Youwei Yan [ctb],
Jinko Graham [aut]

R topics documented:

CEUSNP   *Example data set for LDheatmap*

Description

CEUSNP: Genotypes on 15 SNPs for 60 people

CEUDist: Physical map positions of the 15 SNPs in CEUSNP

Usage

```
data(CEUSNP); data(CEUDist)
```

Format

CEUSNP: A dataframe of SNP genotypes. Each row represents an individual. Each column represents a SNP.

CEUDist: A vector of integers, representing SNP physical map locations on the chromosome.

Details

Data on SNPs with minor allele frequency greater than 5% from a 9kb region of chromosome 7 (base positions 126273659 through 126282556 from release 7 of the International HapMap Project). Genotypes from 30 parent-offspring trios (both parents, one offspring) were obtained. The 30 trios are taken from the so-called CEPH families, a set of multi-generational families from Utah with ancestry from northern and western Europe. From this set of 90 people, 60 parents were extracted.

Source

International HapMap Project ftp://ftp.ncbi.nlm.nih.gov/hapmap/

References

The International HapMap Consortium. A haplotype map of the human genome. Nature 437, 1299-1320. 2005.

Examples

```
data(CEUData)
```

CHBJPTSNP   *Example data set for LDHeatmap*

Description

CHBJPTSNP: Genotypes on 13 SNPs for 45 Chinese and 45 Japanese people CHBJPTDist: Physical map positions of the 13 SNPs

Usage

```
data(CHBJPTSNP); data(CHBJPTDist)
```

Format

CHBJPTSNP: A dataframe of SNP genotypes. Each row represents an individual. Each column represents a SNP.

CHBJPTDist: a vector of integers, representing SNP physical map locations on the chromosome.

Details

The data frame `CHBJPTSNP` contains genotypes for 13 SNPs on chromosome 7, from 45 Chinese and 45 Japanese individuals. The Chinese individuals were unrelated residents of the community at Beijing Normal University with at least 3 Han Chinese grandparents. The Japanese individuals were unrelated residents of the Tokyo metropolitan area with all grandparents from Japan. The data are from release 21 of the International HapMap project (The International HapMap Consortium 2005).

Source

International HapMap Project `ftp://ftp.ncbi.nlm.nih.gov/hapmap/`

References

The International HapMap Consortium. A haplotype map of the human genome. Nature 437, 1299-1320. 2005.

Examples

```
 data(CHBJPTSNP); data(CHBJPTDist)
#Now do our panel plot with LDheatmaps in the panels
library(lattice)
pop<-factor(c(rep("chinese", 45), rep("japanese", 45)))
xyplot(1:nrow(CHBJPTSNP) ~ 1:nrow(CHBJPTSNP) | pop, type="n",
      scales=list(draw=FALSE), xlab="", ylab="",
      panel=function(x, y, subscripts,...) {
        LDheatmap(CHBJPTSNP[subscripts,], CHBJPTDist, newpage=FALSE)})
```

GIMAP5 *Example data set for LDHeatmap*

Description

SNP genotypes on HapMap founders for SNPs spanning the GIMAP5 gene.

Usage

```
data(GIMAP5)
```

Format

GIMAP5 is a list with three elements: snp.data, snp.support and subject.support. snp.data is a `SnpMatrix` object containing the SNP genotypes. Rows correspond to subjects and columns correspond to SNPs. snp.support is a data frame with the following columns:

[,1]	dbSNPalleles	character	alleles at each SNP
[,2]	Assignment	character	same as dbSNPalleles
[,3]	Chromosome	character	chromosome (chr7 for all)
[,4]	Position	numeric	physical position
[,5]	Strand	character	strand (all "+")

subject.support is a one-column data frame with:

[,1]	pop	character	HapMap population of each subject

Details

SNP genotypes from HapMap release 27 for SNPs in a 10KB region spanning the GIMAP5 gene. Data are on founders from each of the 11 HapMap phase III populations:

ASW	African ancestry in Southwest USA
CEU	Utah residents with Northern and Western European ancestry from the CEPH collection
CHB	Han Chinese in Beijing, China

CHD   Chinese in Metropolitan Denver, Colorado
GIH   Gujarati Indians in Houston, Texas
JPT   Japanese in Tokyo, Japan
LWK   Luhya in Webuye, Kenya
MEX   Mexican ancestry in Los Angeles, California
MKK   Maasai in Kinyawa, Kenya
TSI   Toscani in Italia
YRI   Yoruba in Ibadan, Nigeria

Only those SNPs with minor allele frequency greater than 5% in all populations were retained. The base positions are from NCBI build 36 (UCSC genome hg18).

Source

International HapMap Project `ftp://ftp.ncbi.nlm.nih.gov/hapmap/`

References

The International HapMap Consortium. A haplotype map of the human genome. Nature 437, 1299-1320. 2005.

See Also

GIMAP5.CEU

Examples

```
data(GIMAP5)
#Now do a lattice plot with LDheatmaps in the panels
library(lattice)
pop<-GIMAP5$subject.support$pop
n<-nrow(GIMAP5$snp.data)
xyplot(1:n ~ 1:n | pop, type="n", scales=list(draw=FALSE), xlab="", ylab="",
      panel=function(x, y, subscripts,...) {
         LDheatmap(GIMAP5$snp.data[subscripts,],GIMAP5$snp.support$Position,
                   newpage=FALSE)})
rm(pop,n)
```

GIMAP5.CEU   *Example data set for LDheatmap*

Description

SNP genotypes on HapMap founders from the CEU population for SNPs spanning the GIMAP5 gene.

Usage

```
data(GIMAP5.CEU)
```

Format

GIMAP5.CEU is a list with two elements: snp.data and snp.support. snp.data is a `SnpMatrix` object containing the SNP genotypes. Rows correspond to subjects and columns correspond to SNPs. snp.support is a data frame with the following columns:

[,1]	dbSNPalleles	character	alleles at each SNP
[,2]	Assignment	character	same as dbSNPalleles
[,3]	Chromosome	character	chromosome (chr7 for all)
[,4]	Position	numeric	physical position
[,5]	Strand	character	strand (all "+")

Details

SNP genotypes from HapMap release 27 for SNPs in a 10KB region spanning the GIMAP5 gene. Data are on founders from the CEU population, described as Utah residents with Northern and Western European ancestry from the CEPH collection. Only those SNPs with minor allele frequency greater than 5% in all populations were retained. The base positions are from NCBI build 36 (UCSC genome hg18).

Source

International HapMap Project <ftp://ftp.ncbi.nlm.nih.gov/hapmap/>

References

The International HapMap Consortium. A haplotype map of the human genome. Nature 437, 1299-1320. 2005.

See Also

GIMAP5

Examples

```
require(snpStats) # for the SnpMatrix data structure
data(GIMAP5.CEU)
LDheatmap(GIMAP5.CEU$snp.data,GIMAP5.CEU$snp.support$Position)
```

LDheatmap   *This function produces a pairwise LD plot.*

Description

LDheatmap() is used to produce a graphical display, as a heat map, of pairwise linkage disequilibrium (LD) measurements for SNPs. The heat map is a false color image in the upper-left diagonal of a square plot. Optionally, a line parallel to the diagonal of the image indicating the physical or genetic map positions of the SNPs may be added, along with text reporting the total length of the genomic region considered.

Usage

```
LDheatmap(gdat, genetic.distances=NULL, distances="physical",
LDmeasure="r", title="Pairwise LD", add.map=TRUE, add.key=TRUE,
geneMapLocation=0.15, geneMapLabelX=NULL, geneMapLabelY=NULL,
SNP.name=NULL, color=NULL, newpage=TRUE,
name="ldheatmap", vp.name=NULL, pop=FALSE, flip=NULL, text=FALSE)
```

Arguments

gdat
: SNP data: a data frame of genotype objects, a SnpMatrix object, a square matrix of pairwise linkage disequilibrium measurements or an object of class "LDheatmap" (the returned object of this function).

genetic.distances
: A numeric vector of map locations of the SNPs, in the same order as SNPs listed in gdat, in terms of genetic or physical distances. Physical distances should be in bases, genetic distances should be in centiMorgans (cM). When gdat is *not* an object of class LDheatmap, the default is a vector that represents equi-spaced markers, 1kb (1000 bases) apart. When gdat *is* an object of class LDheatmap, the genetic.distances argument is taken to be the genetic.distances list item of gdat.

distances
: A character string to specify whether the provided map locations are in physical or genetic distances. If distances="physical" (default), the text describing the total length of the region will be "Physical Length:XXkb" where XX is the length of the region in kilobases. If distances="genetic", the text will be "Genetic Map Length:YYcM" where YY is the length of the region in centiMorgans. If gdat is an object of class LDheatmap, distances is taken from gdat.

LDmeasure
: A character string specifying the measure of LD - either allelic correlation r^2 or Lewontin's |D′|; default = "r" for r^2; type "D'" for |D′|. This argument is ignored when the user has already supplied calculated LD measurements through gdat (i.e., when gdat is a matrix of pairwise LD measurements or an object of class "LDheatmap").

title
: A character string for the main title of the plot. Default is "Pairwise LD".

add.map	If TRUE (default) a diagonal line indicating the physical or genetic map positions of the SNPs will be added to the plot, along with text indicating the total length of the genetic region.
add.key	If TRUE (default) the color legend is drawn.
geneMapLocation	
	A numeric value specifying the position of the line parallel to the diagonal of the matrix; the larger the value, the farther it lies from the matrix diagonal. Ignored when add.map=FALSE.
geneMapLabelX	A numeric value specifying the x-coordinate of the text indicating the total length of the genomic region being considered. Ignored when add.map=FALSE.
geneMapLabelY	A numeric value specifying the y-coordinate of the text indicating the total length of the genomic region being considered. Ignored when add.map=FALSE.
SNP.name	A vector of character string(s) of SNP name(s) to be labelled. Should match the names of SNPs in the provided object gdat, otherwise nothing is done.
color	A range of colors to be used for drawing the heat map. Default is grey.colors(20).
newpage	If TRUE (default), the heat map will be drawn on a new page.
name	A character string specifying the name of the LDheatmap graphical object (grob) to be produced.
vp.name	A character string specifying the name of the viewport where the heat map is going to be drawn.
pop	If TRUE, the viewport where the heat map is drawn is popped (i.e. removed) from the viewport tree after drawing. Default=FALSE.
flip	If TRUE, the LDheatmap plot is flipped below a horizontal line, in the style of Haploview. Default is FALSE.
text	If TRUE, the LD measurements are printed on each cell.

Details

The input object gdat can be a data frame of genotype objects (a data structure from the **genetics** package), a SnpMatrix object (a data structure from the **snpStats** package), or any square matrix with values between 0 and 1 inclusive. LD computation is much faster for SnpMatrix objects than for genotype objects. In the case of a matrix of LD values between 0 and 1, the values above the diagonal will be plotted. In the display of LD, SNPs appear in the order supplied by the user as the horizontal and vertical coordinates are increased and one moves along the off-diagonal line, from the bottom-left to the top-right corner. To achieve this, the conventions of the image() function have been adopted, in which horizontal coordinates correspond to the rows of the matrix and vertical coordinates correspond to columns, and vertical coordinates are indexed in increasing order from bottom to top. For the argument color, an appropriate color palette for quantitative data is recommended, as outlined in the help page of the brewer.pal() function of the **RColorBrewer** package. See the package vignette LDheatmap for more examples and details of the implementation. Examples of adding "tracks" of genomic annotation above a flipped heatmap are in the package vignette addTracks.

Value

An object of class ″LDheatmap″ which contains the following components:

LDmatrix   The matrix of pairwise LD measurements plotted in the heat map.

LDheatmapGrob   A grid graphical object (grob) representing the produced heat map.

heatmapVP   The viewport in which the heat map is drawn. See viewport.

genetic.distances
                 The vector of the supplied physical or genetic map locations, or the vector of
                 equispaced marker distances when no distance vector is supplied.

distances   A character string specifying whether the provided map distances are physical
                 or genetic.

color   The range of colors used for drawing the heat map.

The grob LDheatmapGrob has three grobs as its children (components). They are listed below
along with their own children and respectively represent the color image with main title, genetic
map and color key of the heat map: ″heatMap″ - ″heatmap″, ″title″; ″geneMap″ - ″diagonal″,
″segments″, ″title″, ″symbols″, ″SNPnames″; and ″Key″ - ″colorKey″, ″title″, ″labels″,
″ticks″, ″box″.

Note

The produced heat map can be modified in two ways. First, it is possible to edit *interactively* the
grob components of the heat map, by using the function grid.edit; the function will not work
if there is no open graphical device showing the heat map. Alternatively, the user can use the
function editGrob and work with the grob LDheatmapGrob returned by LDheatmap. See Examples
for usage. LDheatmap() uses Grid, which does not respond to par() settings. Hence modifying
par() settings of mfrow and mfcol will not work with LDheatmap(). The Examples section shows
how to display multiple heat maps on one plot without the use of par().

Author(s)

Ji-hyung Shin <shin@sfu.ca>, Sigal Blay <sblay@sfu.ca>, Nicholas Lewin-Koh <nikko@hailmail.net>,
Brad McNeney <mcneney@stat.sfu.ca>, Jinko Graham <jgraham@cs.sfu.ca>

References

Shin J-H, Blay S, McNeney B and Graham J (2006). LDheatmap: An R Function for Graphical
Display of Pairwise Linkage Disequilibria Between Single Nucleotide Polymorphisms. Journal of
Statistical Software, **16** Code Snippet 3

See Also

LD, genotype, Grid, LDheatmap.highlight, LDheatmap.marks

Examples

```
# Pass LDheatmap a SnpMatrix object
set.seed(1)
#make an example matrix of genotypes, coded as 0, 1 2 copies of an index allele
gdat<-matrix(rbinom(n=500,size=2,prob=.5),ncol=5)
require(snpStats)
gdat<-as(gdat,"SnpMatrix")
LDheatmap(gdat,genetic.distances=c(0,1000,3000,4000,10000))
#Load the package's data set
data(CEUSNP); data(CEUDist)
#Creates a data frame "CEUSNP" of genotype data and a vector "CEUDist"
#of physical locations of the SNPs
# Produce a heat map in a grey color scheme
MyHeatmap <- LDheatmap(CEUSNP, genetic.distances = CEUDist,
                       color = grey.colors(20))
# Same heatmap, flipped below a horizontal gene map -- for examples of
# adding genomic annotation tracks to a flipped heatmap see
# vignette("addTracks")
# flippedHeatmap<-LDheatmap(MyHeatmap,flip=TRUE)
# Prompt the user before starting a new page of graphics output
# and save the original prompt settings in old.prompt.
old.prompt <- devAskNewPage(ask = TRUE)
# Highlight a certain LD block of interest:
LDheatmap.highlight(MyHeatmap, i = 3, j = 8, col = "black",
fill = "grey",flipOutline=FALSE, crissCross=FALSE)
# Plot a symbol in the center of the pixel which represents LD between
# the fourth and seventh SNPs:
LDheatmap.marks(MyHeatmap,  4,  7,  gp=grid::gpar(cex=2),  pch = "*")
#### Use an RGB pallete for the color scheme ####
rgb.palette <- colorRampPalette(rev(c("blue", "orange", "red")), space = "rgb")
LDheatmap(MyHeatmap, color=rgb.palette(18))
#### Modify the plot by using 'grid.edit' function ####
#Draw a heat map where the SNPs "rs2283092" and "rs6979287" are labelled.
require(grid)
LDheatmap(MyHeatmap, SNP.name = c("rs2283092", "rs6979287"))
#Find the names of the top-level graphical objects (grobs) on the current display
getNames()
#[1] "ldheatmap"
# Find the names of the component grobs of "ldheatmap"
childNames(grid.get("ldheatmap"))
#[1] "heatMap" "geneMap" "Key"
#Find the names of the component grobs of heatMap
childNames(grid.get("heatMap"))
#[1] "heatmap" "title"
#Find the names of the component grobs of geneMap
childNames(grid.get("geneMap"))
#[1] "diagonal" "segments" "title"    "symbols"  "SNPnames"
#Find the names of the component grobs of Key
childNames(grid.get("Key"))
#[1] "colorKey" "title"    "labels"   "ticks"    "box"
#Change the plotting symbols that identify SNPs rs2283092 and rs6979287
#on the plot to bullets
```

```
grid.edit("symbols", pch = 20, gp = gpar(cex = 1))
#Change the color of the main title
grid.edit(gPath("ldheatmap", "heatMap", "title"), gp = gpar(col = "red"))
#Change size of SNP labels
grid.edit(gPath("ldheatmap", "geneMap","SNPnames"), gp = gpar(cex=1.5))
#Add a grid of white lines to the plot to separate pairwise LD measures
grid.edit(gPath("ldheatmap", "heatMap", "heatmap"), gp = gpar(col = "white",
                                                      lwd = 2))
#### Modify a heat map using 'editGrob' function ####
MyHeatmap <- LDheatmap(MyHeatmap, color = grey.colors(20))
new.grob <- editGrob(MyHeatmap$LDheatmapGrob, gPath("geneMap", "segments"),
                     gp=gpar(col="orange"))
##Clear the old graphics object from the display before drawing the modified heat map:
grid::grid.newpage()
grid::grid.draw(new.grob)
# now the colour of line segments connecting the SNP
# positions to the LD heat map has been changed from black to orange.
#### Draw a resized heat map (in a 'blue-to-red' color scale ####
grid::grid.newpage()
grid::pushViewport(viewport(width=0.5, height=0.5))
LDheatmap(MyHeatmap, SNP.name = c("rs2283092", "rs6979287"), newpage=FALSE,
          color="blueToRed")
grid::popViewport()
#### Draw and modify two heat maps on one plot ####
grid::grid.newpage()
##Draw and the first heat map on the left half of the graphics device
grid::pushViewport(viewport(x=0, width=0.5, just="left"))
LD1<-LDheatmap(MyHeatmap, color=grey.colors(20), newpage=FALSE,
               title="Pairwise LD in grey.colors(20)",
               SNP.name="rs6979572", geneMapLabelX=0.6,
               geneMapLabelY=0.4, name="ld1")
grid::upViewport()
##Draw the second heat map on the right half of the graphics device
grid::pushViewport(viewport(x=1,width=0.5,just="right"))
LD2<-LDheatmap(MyHeatmap, newpage=FALSE, title="Pairwise LD in heat.colors(20)",
               SNP.name="rs6979572", geneMapLabelX=0.6, geneMapLabelY=0.4, name="ld2")
grid::upViewport()
##Modify the text size of main title of the first heat map.
grid::grid.edit(gPath("ld1", "heatMap","title"), gp=gpar(cex=1.5))
##Modify the text size and color of the SNP label of the second heat map.
grid::grid.edit(gPath("ld2", "geneMap","SNPnames"), gp=gpar(cex=1.5, col="DarkRed"))
#### Draw a lattice-like plot with heat maps in panels ####
# Load CHBJPTSNP and CHBJPTDist
data(CHBJPTSNP); data(CHBJPTDist)
# Make a variable which indicates Chinese vs. Japanese
pop <- factor(c(rep("chinese",45), rep("japanese",45)))
require(lattice)
xyplot(1:nrow(CHBJPTSNP) ~ 1:nrow(CHBJPTSNP) | pop,
       type="n", scales=list(draw=FALSE), xlab="", ylab="",
       panel=function(x, y, subscripts,...) {
          LDheatmap(CHBJPTSNP[subscripts,], CHBJPTDist, newpage=FALSE) })
data(GIMAP5)
n<-nrow(GIMAP5$snp.data)
```

```
lattice::xyplot(1:n ~ 1:n | GIMAP5$subject.support$pop,
    type="n", scales=list(draw=FALSE), xlab="", ylab="",
    panel=function(x, y, subscripts,...) {
       LDheatmap(GIMAP5$snp.data[subscripts,],
                  GIMAP5$snp.support$Position, SNP.name="rs6598", newpage=FALSE) })
#Reset the user's setting for prompting on the graphics output
#to the original value before running these example commands.
devAskNewPage(old.prompt)
```

LDheatmap.addGenes   *Add gene plot to an LDheatmap object.*

Description

Retrieve genes from the UCSC Genome Browser, generate the genes plot and add it to an LD-heatmap object.

Usage

```
LDheatmap.addGenes(LDheatmap, chromosome, genome = NULL, genesLocation = 0.02,
splice_variants = TRUE, non_coding = TRUE)
```

Arguments

LDheatmap	An object of class LDheatmap.
chromosome	A character string that identifies the chromosome.
genome	The genome assembly to use. The default is the most recent human genome assembly on the UCSC genome browser.
genesLocation	The gene plot distance from the LD heat map gene map.
splice_variants	
	If FALSE, exclude gene splice variants.
non_coding	If FALSE, exclude non-coding genes.

Details

Note: The LDheatmap object should have a non-NULL genetic.distances component. Otherwise the gene map will not be placed correctly. The genes are color coded as follows: black – feature has a corresponding entry in the Protein Data Bank (PDB); dark blue – transcript has been reviewed or validated by either the RefSeq, SwissProt or CCDS staff; medium blue – other RefSeq transcripts; and light blue – non-RefSeq transcripts.

For assemblies older than hg18, all genes are plotted in grey.

Value

An object of class LDheatmap given as an argument, with the grob LDheatmapGrob modified to inclue the "transcripts" child grob.

Author(s)

Sigal Blay <sblay@sfu.ca>

References

<http://genome.ucsc.edu/cgi-bin/hgTrackUi?g=knownGene>

See Also

LDheatmap, plotGenes

Examples

```
data(GIMAP5.CEU)
ll<-LDheatmap(GIMAP5.CEU$snp.data,GIMAP5.CEU$snp.support$Position,flip=TRUE)
# Add gene plot
llplusgenes <- LDheatmap.addGenes(ll, chr="chr7", genome="hg18")
```

LDheatmap.addGrob *Add a graphical object to an LDheatmap plot*

Description

Add a graphical object to an LDheatmap plot such that the x-axis corresponds to the physical map on the heatmap.

Usage

```
LDheatmap.addGrob(LDheatmap, grob, height = 0.2)
```

Arguments

LDheatmap	An object of class LDheatmap.
grob	A graphical object of classgrob.
height	The height of the viewport in which the grob will be placed.

Value

An object of class LDheatmap given as an argument, with the grob LDheatmapGrob modified to inclue the new child grob.

Author(s)

Sigal Blay <sblay@sfu.ca>

See Also

[LDheatmap](#)

Examples

```
# Add an empty rectangle frame
data(GIMAP5.CEU)
ll<-LDheatmap(GIMAP5.CEU$snp.data,GIMAP5.CEU$snp.support$Position,flip=TRUE)
llplusgrob<-LDheatmap.addGrob(ll, grid::rectGrob())
```

LDheatmap.addRecombRate

Add recombination rate plot to an LD heat map.

Description

Retrieve average rates of recombination from the deCODE genetic map from the UCSC Genome
Browser and add them to an LDheatmap object.

Usage

```
LDheatmap.addRecombRate(LDheatmap, chromosome, genome = NULL, recombRateLocation = 0.02,
view = "dense")
```

Arguments

LDheatmap	An object of class LDheatmap.
chromosome	A character string that identifies the chromosome.
genome	The genome assembly to use. The default is the most recent human genome assembly on the UCSC Genome Browser.
recombRateLocation	
	The plot distance from the LD heat map gene map.
view	Display mode. Possible values are "dense" (the default), "squish", "pack" and "full".

Value

An object of class LDheatmap given as an argument, with the grob LDheatmapGrob modified to
inclue the "recombRate" child grob.

Author(s)

Sigal Blay <sblay@sfu.ca>

References

<http://genome.ucsc.edu/cgi-bin/hgTrackUi?g=recombRate>

See Also

[LDheatmap](), [recombRate]()

Examples

```
data(GIMAP5.CEU)
ll<-LDheatmap(GIMAP5.CEU$snp.data,GIMAP5.CEU$snp.support$Position,flip=TRUE)
# Add recombination rate plot
ll_recomb <- LDheatmap.addRecombRate(ll, chr="chr7", genome="hg18")
```

LDheatmap.addScatterplot

Add a scatter plot to an LDheatmap object

Description

Add a scatter plot to an LDheatmap object. The x axis is the map of genetic distances of the SNPs.

Usage

```
LDheatmap.addScatterplot(LDheatmap, P, height = 0.2, ylab = NULL, ylim=NULL,
type = "points")
```

Arguments

LDheatmap	An object of class LDheatmap.
P	A vector with the values to be plotted as the y axis.
height	The height of the plot.
ylab	The y axis label.
ylim	The y axis limits.
type	Plot type. Possible values are "points" (the default), "lines" or "both".

Details

The function creates an "association" grob and adds it to the LDheatmap object. Then it pushes a viewport and draws the LDheatmapGrob onto it.

Value

An object of class LDheatmap given as an argument, with the grob LDheatmapGrob modified to inclue the "association" child grob.

Note

The function can display at most two scatter plots in the default setting. To add three or more scatter plots in the same viewport, the user can change the input value "location" from function constructVP which is the function inside the LDheatmap.addScatterplot. The default "location" value is 0.03, for adding the third scatter plot, user needs to set the "location" to 0.23, where 0.2 units is the height of the scatter plot. For the fourth scatter plot, set the "location" to 0.43 etc. See Examples for usage.

Author(s)

Sigal Blay <sblay@sfu.ca> and more

See Also

LDheatmap

Examples

```
# Load the package's data set
data("CEUSNP"); data("CEUDist")
# Produce an LDheatmap object
MyLDheatmap <- LDheatmap(CEUSNP, genetic.distances = CEUDist, flip = TRUE)
# Generate an arbitrary vector of values to plot
Yvalues <- seq(length = length(MyLDheatmap$genetic.distances), from = 0.01, to = 0.5)
# Add scatter plot
assoc <- LDheatmap.addScatterplot(MyLDheatmap, Yvalues)
######## Adding three or more scatter plots ########
# Redefine LDheatmap.addScatterplot() to display the third scatter plot
LDheatmap.addScatterplot_test3 <- function(LDheatmap, P, height=0.2, ylab=NULL,
ylim=NULL, type="points",color,pch) {
if (dim(LDheatmap$LDmatrix)[1] != length(P)) {
print("Length of vector not equal number of SNPs in LDheatmap")
return()

flip <- !is.null(LDheatmap$flipVP)
vp <- constructVP(LDheatmap$LDheatmapGrob, 0.23, flip)
......
return(LDheatmap)
}}
environment(LDheatmap.addScatterplot_test3) <- asNamespace('LDheatmap')
```

LDheatmap.highlight   *Highlight a genetic region in the linkage disequilibrium heat map*

Description

The function LDheatmap.highlight() is used to highlight a specified genetic region in the linkage disequilibrium (LD) heat map drawn with the LDheatmap() function.

Usage

```
LDheatmap.highlight(LDheatmap, i, j, fill = "NA", col = "black", lwd = 1,
lty = 1,flipOutline=FALSE, crissCross = FALSE)
```

Arguments

LDheatmap   An object of class "LDheatmap" returned by the function LDheatmap().

i   A numeric value specifying the index of the first SNP to be in the highlighted
              region.

j   A numeric value specifying the index of the last SNP, which must be different
              from i, to be in the highlighted region.

fill   Color to fill the highlighted area with.

col   A character string specifying the color of the line segments defining the bound-
              ary of highlighted region; see [par](() for possible values.

lwd   A *positive* number specifying the width of the boundary segments.

lty   Either an integer or a character string specifying the line type of the boundary
              segments; see [par](() for possible values.

flipOutline   A Boolean variable that flips the outlined section over the diagonal of the heatmap.

crissCross   A Boolean variable that controls whether a contiguous selection of SNPs are
              outlined only on their polygonal boundary or at individual SNP levels.

Value

A data frame of the x and y coordinates of points defining the border of the highlighted area.

Warning

By default, LDheatmap.highlight() finds the viewport to draw on from the LDheatmap object
passed to it as an argument. However, if LDheatmap() was called with the option pop=TRUE, the
resulting LDheatmap object is not assigned a viewport. In this case, LDheatmap.highlight()
assumes the user wishes to highlight in the current viewport. Therefore, if LDheatmap() has been
called with the option pop=TRUE, the user must navigate to the correct viewport before calling
LDheatmap.highlight().

Note

The function LDheatmap.highlight() highlights the cells representing the pairwise LD for the
SNPs located between i-th and j-th (inclusive) SNPs in the genomic region of interest. The order
of indices has no effect on the plot. For example, LDheatmap.highlight(LDheatmap, i=2, j=4)
is the same as LDheatmap.highlight(LDheatmap, i=4, j=2), which highlights the cells repre-
senting the pairwise LD for the second, third and fourth SNPs.

Author(s)

Nicholas Lewin-Koh <nikko@hailmail.net>, Ji-Hyung Shin <shin@sfu.ca>, Sigal Blay <sblay@sfu.ca>

Examples

```
data(CEUSNP); data(CEUDist)
tt <- LDheatmap(CEUSNP, genetic.distances=CEUDist)
LDheatmap.highlight(tt, 3, 8, col="blue", fill="green", lwd=3, flipOutline=FALSE, crissCross=FALSE)
```

LDheatmap.marks   *Plots a symbol in the centers of cells of the heat map image*

Description

The function LDheatmap.marks() is used to plot a symbol in the centers of cells representing the pairwise linkage disequilibria of specified pairs of SNPs.

Usage

```
LDheatmap.marks(LDheatmap, i, j = NULL, pch = 20, gp=gpar(...), ...)
```

Arguments

LDheatmap	An object of class "LDheatmap" returned by the function [LDheatmap]().
i	A vector of indices of the first set of SNPs.
j	A vector of indices of the second set of SNPs.
pch	Either an integer value or a single character specifying the symbol to be plotted. See [points]() for possible values and their corresponding symbols.
gp	Graphical parameters; See [gpar]().
...	Graphical parameter settings to be passed on to the [gpar]() function.

Details

The lengths of the vectors i and j must be the same and greater than or equal to 1. If the lengths are greater than 1, the function plots the specified symbol in the centers of the (i^k, j^k)-th cells (for k=1,...K; K = length of the vectors i and j), where i^k and j^k are the k-th elements of vectors i and j, respectively. For example, if i=c(1,2) and j=c(3,5), LDheatmap() plots a symbol in the centers of the cells representing pairwise linkage disequilibria between the first and third SNPs and between the second and fifth SNPs in the genome of interest. Note that the order of the sets of indices does not matter; for example, LDheatmap.marks(LDheatmap, i=c(1,2), j=c(3,5)) is equivalent to LDheatmap.marks(LDheatmap, i=c(3,5), j=c(1,2)).

Value

x	The vector of x coordinate(s) of the plotted symbol(s).
y	The vector of y coordinate(s) of the plotted symbol(s).

Warning

By default, LDheatmap.marks() finds the viewport to draw on from the LDheatmap object passed to it as an argument. However, if LDheatmap() was called with the option pop=TRUE, the resulting LDheatmap object is not assigned a viewport. In this case, LDheatmap.marks() assumes the user wishes to highlight in the current viewport. Therefore, if LDheatmap() has been called with the option pop=TRUE, the user must navigate to the correct viewport before calling LDheatmap.marks().

Author(s)

Nicholas Lewin-Koh <nikko@hailmail.net>, Ji-Hyung Shin <shin@sfu.ca>, Sigal Blay <sblay@sfu.ca>

Examples

```
data(CEUSNP); data(CEUDist)
tt <- LDheatmap(CEUSNP, genetic.distances=CEUDist)
LDheatmap.marks(tt, 15, 3, cex=1.6, col="blue")
```

---

plotGenes	*Plot genes from a specified region of the human genome.*

Description

Retrieves genes from the UCSC Genome Browser and generate the genes plot.

Usage

```
plotGenes(minRange, maxRange, chromosome, genome = "hg19", plot_lines_distance = 0.03,
vp = viewport(x = 0, y = 0.99, just = c("left", "top")), splice_variants = TRUE,
non_coding = TRUE)
```

Arguments

minRange   The sequence minimum range in base pairs.

maxRange   The sequence maximum range in base pairs.

chromosome   A character string identifying the chromosome.

genome   The genome assembly to use. The default is hg19, the most recent human genome assembly on the UCSC genome browser.

plot_lines_distance

                The distance between the lines of genes plotted.

vp   A viewport.

splice_variants

                If FALSE, exclude gene splice variants.

non_coding   If FALSE, exclude non-coding genes.

Details

The genes are color coded as follows: Black – feature has a corresponding entry in the Protein Data Bank (PDB) Dark blue – transcript has been reviewed or validated by either the RefSeq, SwissProt or CCDS staff Medium blue – other RefSeq transcripts Light blue – non-RefSeq transcripts

For assemblies older than hg18, all genes are plotted in grey.

Value

A grob of gene plots.

Author(s)

Sigal Blay <sblay@sfu.ca> and more

References

http://genome.ucsc.edu/cgi-bin/hgTrackUi?g=knownGene

Examples

```
grid::grid.newpage()
plotGenes(149500000, 150000000, "chr7")
```

---

recombRate   *Produce recombination rate plot.*

Description

Plot average rates of recombination from the deCODE genetic map for a specified genetic sequence.

Usage

```
recombRate(minRange, maxRange, chromosome, genome = "hg19", vp = viewport(x = 0,
y = 0.99, height = 0.04, just = c("left", "top")), view = "dense")
```

Arguments

minRange	The sequence minimum range in base pairs.
maxRange	The sequence maximum range in base pairs.
chromosome	A character string identifying the chromosome.
genome	The genome assembly to use. The default is hg19, the most recent human genome assembly on the UCSC genome browser.
vp	A viewport.
view	Display mode. Possible values are "dense" (the default), "squish", "pack" and "full".

Value

A grob representing recombination rates.

Author(s)

Sigal Blay <sblay@sfu.ca> and more

References

<http://genome.ucsc.edu/cgi-bin/hgTrackUi?g=recombRate>

Examples

```
grid::grid.newpage()
recombRate(129000000, 140000000, "chr7", "hg18")
grid::grid.newpage()
grid::pushViewport(grid::viewport(width=0.8, x=0.2, just="left"))
recombRate(129000000, 140000000, "chr7", "hg18", view="full")
grid::popViewport()
```

---

vcfR2SnpMatrix   *Extract genotype information from a vcfR object*

Description

Convert a `vcfR-class` object into a list composed of genetic distances, subject IDs, and a `SnpMatrix-class`/`XSnpMatrix-class` object.

Usage

```
vcfR2SnpMatrix(obj, phased = NULL, subjects = NULL)
```

Arguments

obj   An instance or path of the `vcfR-class` object to be processed.

phased   If TRUE the output genotype data are in the form of a `XSnpMatrix-class` object.
             Otherwise, they are in the form of a `SnpMatrix-class` object. If it is unspeci-
             fied, the phasing status will be determined by checking the first entry in the gt
             slot of the vcfR object. If the first entry is also missing, the value will be set to
             FALSE.

subjects   A character or factor containing subject IDs. If supplied, genotype info of only
             those subjects will be returned. This should be a subset of the sample IDs in the
             vcfR object.

Details

In order to let vcfR2SnpMatrix function properly, the input vcfR-class object is expected to be generated from a valid VCF file which contains only biallelic SNPs and includes a GT section.

Value

A list which contains the following components:

genetic.distances

A numeric vector of the reference positions of SNPs.

subjectID   A character vector of IDs of the subjects which the returned genotype data belong to.

data   An object of SnpMatrix-class/XSnpMatrix-class containing genotype data.

See Also

io_vcfR, SnpMatrix-class, XSnpMatrix-class

Examples

```
# Load the vcfR object -- requires vcfR
 if (requireNamespace("vcfR", quietly = TRUE)) {
   require(vcfR)
   data(vcfR_example)
   vcf <- vcf[8:12,]
   # Extract needed genotype information
   alist <- vcfR2SnpMatrix(vcf)

   # Draw a pairwise LD plot using the extracted data
   LDheatmap(alist$data, alist$genetic.distance)
 }
```

Index