idiogram

November 11, 2009

R topics documented:

`buildChromLocation.2`
A function to generate an instantiation of a chromLocation class

---

Description

This function will take the name of a data package and build a chromLocation object representing that data set. It has also been modified to allow further breakup of the chromLocs.

Usage

```
buildChromLocation.2(dataPkg,major=NULL)
```

Arguments

`dataPkg`	The name of the data package to be used
`major`	name of major breakpoint by which to divide chromosomes, "arms", "bands", and "mb" currently work.

Details

The requested data set must be available in the user's `.libPaths()`, and the function will throw an error if this is not the case.

If the data package is present, the necessary information will be extracted from the data package and a `chromLocation` object will be created.

If "major" is set to "arms", the the chromLocs object is populated with data from the chromosome arms; "1p", "1q", "2p", etc... Rat and Human chromsomes follow this pattern , so data packages from both species should work with this function.

If "major" is set to "bands", the chromosomes are divided up based upon which band they fall into.

If "major" is set to "mb", chromosomes are split into 3000+ megabase segments. Note, this creates a very large chromLocation object.

Note, "major" can contain multiple breakpoint names, eg. major=c("arms","bands")

If the "major" argument is used, it stores a list of the extra chromosome names. `chromLoc@chromLocs$armList` - (or `bandList`, `mbList`)

Value

A `chromLocation` object representing the specified data set.

Author(s)

Main author: Jeff Gentry with minor additions by: Karl Dykema

Examples

```
## A bit of a hack to not have a package dependency on hgu95av2
## but need to fiddle w/ the warn level to not fail the example anyways.
curWarn <- getOption("warn")
options(warn=0)
on.exit(options(warn=curWarn), add=TRUE)
if (require(hgu95av2) & require(idiogram)) {
  data(Hs.cytoband)
  z <- buildChromLocation.2("hgu95av2",major="arms")
} else print("This example requires the hgu95av2 data package")
```

---

cytoband-class *Class "cytoband"*

Description

Cytogenetic banding information

Objects from the Class

Objects can be created by calls of the form `new("cytoband", ...)`.

Slots

stain: typical staining designation from classical cytogenetics

band: character string representing the band name/number

start: position in basepairs for the start of a given band

end: position in basepairs for the end of a given band

length: length in basepairs for a given band

Author(s)

Karl Dykema <karl.dykema@vai.org>

See Also

See Also as `Hs.cytoband`

Examples

```
## None
```

`Hs.cytoband` *Cytogenetic Banding information*

Description

Cytogenetic banding information for Homo sapiens (Hs) ,Mus musculus (Mm) and Rattus norvegicus (Rn)

Usage

```
data(Hs.cytoband)
```

Format

Hs.cytoband is an environment containing 24 objects of class "cytoband" as defined by the idiogram package. Rn.cytoband and Mm.cytoband have also been included for the rat and mouse genomes.

Source

UCSC Genome Browser http://genome.ucsc.edu/

Examples

```
data(Hs.cytoband)
cyto <- get("1",env=Hs.cytoband)
bands <- matrix(cyto@end-cyto@start,ncol=1)
barplot(bands,col="white")
```

idiograb *idiograb*

Description

idiograb reads the position of the graphics pointer when the (first) mouse button is pressed. idiogram also reads a *second* position of the graphics pointer after another mouse button press. The two points selected are used to define a diagonal line from which a bounding box will be constructed. It then retrieves the gene identifiers of the points that lie within the bounding box.

Usage

```
idiograb(idio, show.box = TRUE, brush = NULL, ...)
```

Arguments

idio	point coordinates and cooresponding gene identifiers from an idiogram function call
show.box	boolean. if TRUE, a box is drawn showing the selected region
brush	a color to highlight the points within the selected region
...	additional plotting paramters passed to points to modify the points within the selected region

Details

Coordinates can be passed in a plotting structure (a list with x, y, and labels components). Typically this is generated from a call to idiogram.

The points selected are used to define the top-left and bottom-right locations *or* the bottom-left and top-right locations for bounding box. These locations can be selected in any order. A character vector of the labels of all the points that lie within the selected region is returned.

Value

A character vector of gene identifiers

Author(s)

Karl Dykema <karl.dykema@vai.org>

See Also

idiogram

Examples

```
data(idiogramExample)
ip <- idiogram(colo.eset[,1],ucsf.chr,chr="1")
if(interactive()) idiograb(ip,brush="red")
```

idiogramExample *data included for idiogram package examples*

Description

colo.eset and ucsf.chr are included for use in example plots. vai.chr is a chromLocation object included for examples in the package 'reb'. Please see the aCGH Bioconductor package for more information.

Usage

```
data(idiogramExample)
```

Source

http://www.bioconductor.org/repository/devel/package/html/aCGH.html

Examples

```
library(idiogram)
data(Hs.cytoband)
data(idiogramExample)
idiogram(colo.eset,ucsf.chr,chr="1")
```

idiogram *Plotting of Genomic Data*

Description

Function for plotting genomic data along with corresponding cytogenetic banding information

Usage

```
idiogram(data, genome, chr=NULL, organism=NULL,
method=c("plot","matplot","image"), margin=c("ticks","idiogram"),
grid.col=c("red","grey"), grid.lty=c(1,2), widths=c(1,2),
relative=FALSE, dlim=NULL, main=NA, xlab=NA,
ylab=NA,cex.axis=.7,na.color = par("bg"),cen.color="red",...)
```

Arguments

data	a vector or matrix of numeric data to plot. The names/rownames attribute *needs* to contain corresponding gene identifiers
genome	a chromLocation object associated with the specified data set. See below for details.
chr	which chromosome to plot
organism	if NULL, determination of the host organism will be retrieved from the organism slot of the chromLocation object. Otherwise "h", "r", or "m" can be used to specify **h**uman, **r**at, or **m**ouse chromosome information

method	plotting method
margin	type of banding information to display in the plot margin
grid.col	a two element vector specifying the centromere and band grid colors.
grid.lty	a two element vector specifying the centromere and band grid line type.
widths	a two element vector specifying the relative width of the margin idiogram two the adjacent graph. This option is currently ignored.
relative	If relative is TRUE, the vertical height of the plot is scaled relative to the size of largest chromosome.
dlim	a two element vector specifying the minimum and maximum values for x. Data in x that exceed the min and max limits will be set to the min/max limit values before plotting.
main	an overall title for the plot. Defaults to the chromosome name.
xlab	a title for the x axis.
ylab	a title for the y axis.
cex.axis	the magnification to be used for axis annotation relative to the current.
na.color	color to be used for NA values, defaults to: par("bg")
cen.color	color to be used for the centromere when margin="idiogram", defaults to: "red"
...	additional graphical parameters can be given as arguments.

Details

This function displays cytogenetic banding information in the plot margin and calls a secondary plotting function to display associated data at the same relative position. Cytogenetic data for human, mouse, and rat genomes are currently included.

The data is arranged by associating gene identifiers to genomic location using a chromLoc annotation object built using the buildChromLocation function from the annotation package. As such, a vector of data is to be plotted, the names attribute of the vector *needs* to contain the gene identifiers. Likewise if a matrix of data is to be plotted, the rownames attribute of the matrix *needs* to contain the gene identifiers.

To date, plot can be called for vector data, while matplot and image can be called for matrix data. Most additional plotting arguments can be passed down via .... However, the idiogram function plots the axis independently. Currently, only the cex.axis,col.axis, and font.axis parameters are intercepted from ...and redirected to the specialized axis call. Other parameters that effect the axis should be set via par.

The function midiogram is a simple wrapper around idiogram to plot **all the chromosomes from a particular organism** using sensible default values. The "m" refers to plotting **m**ultiple idiograms.

Author(s)

Kyle Furge <kyle.furge@vai.org> and Karl Dykema <karl.dykema@vai.org>

See Also

buildChromLocation, Hs.cytoband, idiograb

Examples

```
library(idiogram)

##
## NOTE:This requires an annotation package to work.
##      In this example packages "hu6800.db" and "golubEsets" are used.
##      They can be downloaded from http://www.bioconductor.org
##      "hu6800.db" is under MetaData, "golubEsets" is under Experimental
##      Data.

if(require(hu6800.db) && require(golubEsets)) {
   library(golubEsets)
   data(Golub_Train)

   hu.chr <- buildChromLocation("hu6800")
   ex <- assayData(Golub_Train)$exprs[,1]

   ## make sure the names() attribute is set correctly
   gN <- names(ex)
   gN[1:10]

   idiogram(ex,hu.chr,chr="1")

   colors <- rep("black",times=length(ex))
   colors[ex > 10000] <- "red"
   pts <- rep(1,times=length(ex))
   pts[ex > 10000] <- 2
   idiogram(ex,hu.chr,chr="1",col=colors,pch=pts,font.axis=2,cex.axis=1)
   abline(v=0,col="darkgreen")

   ## An example of the dlim option. It is most useful for making
   ## consistant multi-panel plots
   colors <- rep("black",times=length(ex))
   colors[ex > 10000] <- "red"
   colors[ex < 0] <- "blue"

   idiogram(ex,hu.chr,chr="1",col=colors,xlim=c(-3000,21000))
   idiogram(ex,hu.chr,chr="1",col=colors,dlim=c(-100,7500),xlim=c(-3000,21000))
   idiogram(ex,hu.chr,chr="1",col=colors,dlim=c(-100,7500),xlim=c(-3000,10000))

   ## Using the identify function
   ip <- idiogram(ex,hu.chr,chr="1",col=colors,pch=19)
   #identify(ip$x,ip$y,labels=ip$labels)

} else print("This example requires the hu6800.db and golubEsets data packages.")

## The example data is BAC array CGH data from J. Fridlyand's
## aCGH package

data(idiogramExample)

idiogram(colo.eset[,1],ucsf.chr,chr="1")

idiogram(colo.eset,ucsf.chr,chr="1",method="image")

idiogram(colo.eset,ucsf.chr,chr="1",method="image",col=topo.colors(50),grid.lty=c(1,NA))
```

```
idiogram(colo.eset,ucsf.chr,chr="1",method="mat",type="l")

## for a consistant multi-panel plot it can be helpful to force the data
## range within each panel to a defined range using 'dlim'
## This is similar to calling the 'midiogram' function

op <- par(no.readonly=TRUE)
par(mai=par("mai")*c(0.1,0.5,0.5,0.5))
layout(rbind(c(1:8),c(0,9:14,0),c(15:22)))

for(i in c(1:22)) {
  idiogram(colo.eset,ucsf.chr,chr=i,method="i",dlim=c(-1,1),margin="i",relative=TRUE)
}

par(op)
```

Index