Package 'iGenomicViewer'

April 6, 2012

Version 2.4.8

Date Apr 05, 2012

Title Tool for sending interactive bioinformatic heatmaps with tool-tip content.

Author Daniel P Gaile <dpgaile@buffalo.edu>, Lori A. Shepherd <las65@buffalo.edu>, Lara Sucheston <lsuchest@buffalo.edu>,Andrew Bruno <aebruno2@ccr.buffalo.edu>, Kenneth F. Manly <manly@buffalo.edu>

Maintainer Lori A. Shepherd <las65@buffalo.edu>

Depends R (>= 2.10), sendplot (>= 3.8.0)

Suggests SNPchip, aCGH

biocViews Software,Visualization,DataRepresentation,MultipleComparisons,Bioinformatics,Infrastructure,Genetics,Annotation,Assa

Description A tool for visualizing data

License GPL (>= 2)

URL http://sphhp.buffalo.edu/biostat/research/software/igenomicviewer/index.php

Repository CRAN

Date/Publication 2012-04-06 12:39:22

R topics documented:

iGenomicViewer-package 2
anninfo-class 3
annObj 4
annotation 4
Band.Info 7
bandinfo-class 8

CancerGenes . . . 9
convertLoc . . . 9
cytoBand . . . 11
DiseaseGenes . . . 12
DNArepairgenes . . . 13
GGVobj-class . . . 13
HB19Kv2.HG18 . . . 15
iGGV . . . 16
iGGVex . . . 21
iGGVtiled . . . 21
initGGV . . . 26
initTile . . . 30
makeBandInfo . . . 32
makeGGV . . . 34
makeTiled . . . 37
makeTrack . . . 38
mapobj-class . . . 40
mapping.info . . . 40
mappingObj . . . 41
TIplot-class . . . 45
updateGGV . . . 46
writeExFiles . . . 48

Index **49**

`iGenomicViewer-package`

Wrapper Package to make Bioinformatics heatmaps utilizing sendplot

---

Description

This package creates interactive bioinformatics heatmaps across a genomic range. It allows for the ability to create interactive annotation tracks and an interactive statistical or descriptive plot along side the heatmap.

Author(s)

Lori A. Shepherd

References

Albertson, D.G., Pinkel, D. 2003 Genomic microarrays in human genetic disease and cancer. 12. Spec No 2: 142-152

Chari, R., Lockwood, W.W., Lam, W.L. 2006 Computational Methods for the Analysis of Array Comparative Genomic Hybridization. Cancer Informatics 2: 48-58

Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D. 1998. Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. USA. 95:14863-14868

Gaile, D.P., Shepherd, L.A., Sucheston, L., Bruno, A., Manly, K.F. 2009. sendplot: Took for sending interactive plots with tool-tip content. http://cran.r-project.org/

Hibbs, M.A., Dirksen, N.C., Li, K., Troyanskaya, O.G. 2005. Visualization methods for statistical analysis of microarray clusters. BMC Bioinformatics 6: 115

Yee Hwa Yang, Paquet, A., Dudoit, S. 2007. marray: Exploratory analysis for two-color spotted microarray data R package version 1.20.0 http://www.maths.usyd.edu.au/u/jeany/

Prasad, T.V., Ahson, S.I. 2006. Visualization of microarray gene expression data. Bioinformation 4: 141-145

Zorn, W 2007 JavaScript, DHTML Tooltips JavaScript Cross Browser Library http://www.walterzorn.com/tooltip/tooltip_e.h

See Also

iGGV, iGGVtiled

`anninfo-class` *anninfo 'annotation information' class*

Description

The anninfo object contains all information for a given annoation set. This includes but is not limited to names, chromosome locations, and genomic locations.

anninfo object structure

annotation a data frame of information for the annotation set. The information included depends on the annotation file used and the user settings when creating the object with the makeAnnotation function. The data.frame must have Label and Chrom, as well as one of the options for genomic locations. see `makeAnnotation` for more details.

links a data frame of web address information for the annotation set. It should have the same row dimension as the annotation data.frame.

images a data frame of paths to images for the annotation set. It should have the same row dimensions as the annotation data.frame

Author(s)

Lori A. Shepherd

See Also

annotation

annObj *Default Object of class annObj*

Description

This dataset contains an object of the class annObj. The annObj class consists of anninfo objects which provide annotation information.

Usage

```
data(annObj)
```

Format

see annotation for more details

Details

This dataset contains an object of the class annobj. An annobj contains individual objects of the class anninfo. Each anninfo object is an annotation set. The annotation sets included in this example are disease genes, cancer genes, and DNA repair genes. Other custom annotation tracks may be included at anytime. see anninfo for more information on this class. see CancerGenes, DiseaseGenes, and DNArepairGenes for details on the files used to create annotation tracks.

annotation *Functions for Creating the Annotation Object*

Description

These functions are designed to create an annotation object.

Usage

```
makeAnnotation(file,
               label,
               chrom,
               chrom.levels,
               band.info=NA,
               loc=NA,
               loc.start=NA,
               loc.stop=NA,
               file.sep="\t",
               additional=NA,
               names.additional = NA,
               links=NA,
```

```
                names.links=NA,
                images=NA,
                names.images=NA,
                returnVl = TRUE,
                saveVl = FALSE,
                saveName="Annotation.RData",
                ...)


annotationObj(annotation,
              annotationObj = NA,
              obj.name = NA,
              returnVl = TRUE,
              saveVl = FALSE,
              saveName="AnnotationObj.RData")
```

Arguments

file	path name of file
label	column in file that represents the label for annotation region
chrom	column in file that represents the chromosome location
chrom.levels	vector indicating how the chrom column is represented. i.e chr1, chrom1, 1, etc.
band.info	a band.info object. see makeBandInfo
loc	column in file representing genomic location. See details for more information
loc.start	column in file representing starting genomic location. See details for more information
loc.stop	column in file representing ending genomic location. See details for more information
file.sep	seperation character for file
additional	indication of additional columns in file that should be included in annotation object
names.additional	optional header names for additional columns to be included
links	indication of columns in file that represent hyperlinks, character or numeric, or a data.frame of hyperlinks, with rows equal to the number of rows in the file, and in the same order as they appear in the file
names.links	optional header names for links
images	indication of columns in file that represent images, character or numeric, or a data.frame, with rows equal to the number of rows of the file, of images that must be in the same order as they appear in the file
names.images	optional header names for images
returnVl	logical indicating if object created should be returned
saveVl	logical indicating if object created should be saved to a file

saveName	if saveVl, path name of file
annotation	annotation object created by the makeAnnotation function
annotationObj	annotation object created by annotationObj function. If NA, creates a new object.
obj.name	name of the annotation set in the annotationObj
...	additional arguments to the read.table function for reading the file

Details

The annotationObj is a larger object that contains all individual annotation information desired for a genomic mapping. These annotations will be represented in a track along side of the main heatmap of the iGGV function. Each individual annotation information object must be added separatly to the main annotationObj.

The annotation file must minimally contain columns for name, chromosome, and genomic location. The genomic location may be given by a single value assumed to be a central location and should be specified using the loc argument. This value should be the genomic location with respect to the entire genome not the location within a chromosome. The other, recommended, way of specifing the genomic location is with both a starting and stopping genomic location, and should be specified using loc.start and loc.stop arguments. These values are the location within the chromosome not over the entire genome. If loc.start and loc.stop are used, loc should be NA.

The band.info argument is an object of the class band.info. This object contains information about chromosome, arm, broad.band, and fine.band regions. See band.info help files for details.

The iGGV function has the capabilities of including hyperlinks in displayed tool-tips. In order for the links to display properly they need to be specified in the annotation object. If there are links in the file, the column number or head name may be specified as links. The user may add their own hyperlinks by assigning the links argument to a table of the dimension n by m where n is equivalent to the number of items in the original file. The links must be in proper format of a valid web address, i.e http://completewebpathhere.com . Similarly, images may also be displayed and set through the images argument.

Value

annotationObj returns an object containing annotation information objects.

makeAnnotation returns an object of the class anninfo. This object contains annotation information such as labels, chromosome locations, genomic locations, and web links.

Note

For an example object see annObj

Author(s)

Lori A. Shepherd

See Also

anninfo, annObj,bandinfo, iGGV

Examples

```
library("iGenomicViewer")

#
# The following code will build the same object as data(annObj)
#


# writes out text files to use
writeExFiles()

# loads a band.info object
data(Band.Info)

# makes anninfo object for cancerGenes
annotation1 = makeAnnotation(file="CancerGenes.txt", file.sep="\t", label=2, chrom=3,chrom.levels=c("chr1","chr2

# makes anninfo object for DiseaseGenes
annotation2 = makeAnnotation(file="DiseaseGenes.txt", file.sep="\t", label=2, chrom=3,chrom.levels=c("chr1","chr


# makes anninfo object for DNArepairGenes
annotation3 = makeAnnotation(file="DNArepairgenes.txt", file.sep="\t", label=1, chrom=2, chrom.levels=c("chr1","

# take individual anninfo objects and create a master annotation object
annObj = annotationObj(annotation1, obj.name="CancerGenes")
annObj = annotationObj(annotation2, annotationObj=annObj, obj.name="DiseaseGenes")
annObj = annotationObj(annotation3, annotationObj=annObj, obj.name="DNArepairGenes")
```

`Band.Info` *Default Object of Class Band.Info*

Description

This dataset contains an object of the class band.info that is used as the default band.info object for iGGV functions.

Usage

```
data(Band.Info)
```

Format

see `bandinfo` for more detail

Details

This dataset contains an object of the class band.info that is used as the default band.info object for iGGV functions. It is created using the cytoband.txt file downloaded from the UCSC Genome Bioinformatics website. See also `bandinfo`, `makeBandInfo`, `cytoBand`

Note

This object was created with the cytoband file from the UCSC website

Source

UCSC Genome Bioinformatics http://hgdownload.cse.ucsc.edu/goldenPath/hg18/database/cytoBand.txt.gz

References

UCSC Genome Bioinformatics http://hgdownload.cse.ucsc.edu/goldenPath/hg18/database/cytoBand.txt.gz

---

bandinfo-class *bandinfo 'band information' object*

Description

The bandinfo object contains useful data.frames of starting and ending genomic locations for chromosomes, arms, broad.bands, and fine.bands. Each data.frame has five columns: the region (chrom, arm), label, lower, center, and upper. lower, center, and upper refer to genomic locations. label is the label that will be used for the region when plotting. The bandinfo object will also contain an offset object. This is a numeric vector of length equal to the number of chromsomes. It contains the numeric buffer that should be added to genomic locations given within chromosome to get the location with respect to the entire genome.

mapobj object structure

offset offset is a numeric vector containing buffers to convert genomic locations with respect to chromosome to genomic locations with respect to the entire genome

Chrom A data.frame with 5 columns: Chrom, Label, lower, center, and upper

Arm A data.frame with 5 columns: Arm, Label, lower, center, and upper

Broad.Band A data.frame with 5 columns: Broad.Band, Label, lower, center,and upper

Fine.Band A data.frame with 5 columns: Fine.Band, Label, lower, center,and upper

Author(s)

Lori A. Shepherd

See Also

`makeBandInfo`, `Band.Info`, `cytoBand`

CancerGenes *Cancer Gene Annotation File*

Description

This file is an annotation file for known cancer genes. It has been altered slightly to include hyperlinks to the UCSC Genome Browser.

Usage

```
data(CancerGenes)
```

Format

tab delimited text file with columns for gene name, chromosome, start location, end location, and weblink to UCSC Genome Browser.

Details

annotation files must minimally have columns for gene, chromosome, start location, and end location. This file was altered to also include weblinks to the human genome browser at UCSC. see also `annotation` and `anninfo`

Note

see also `annotation`, `anninfo`, `annObj`

References

Futreal PA, Coin L, Marshall M, Down T, Hubbard T, Wooster R, Rahman N, Stratton MR. A census of human cancer genes. Nat Rev Cancer. 2004 Mar;4(3):177-83

Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D. The human genome browser at UCSC. Genome Res. 2002 Jun;12(6):996-1006

convertLoc *Converts Genomic Locations*

Description

This function will convert genomic locations with respect to chromosome to genomic locations with respect to the entire genome or vice versa.

Usage

```
convertCloc(x,
            chr,
            row=TRUE,
            bandobj=NA)

convertGloc(x,
            chr,
            row=TRUE,
            bandobj=NA)
```

Arguments

x	numeric vector or matrix of values
chr	indication of chromosome
row	when x is a matrix, should the chr value be applied by row or by column
bandobj	an object of the class bandinfo. if NA, useds iGenomicViewer's default band.info object

Details

Some applications require genomic locations to be with respect to the genome instead of by chromosome and vice versa. These functions provide a way to convert a numeric vector or matrix of one type to the other.

chr indicates which chromosome the value is on. If x is a vector, chr should be a vector of equal length, or a single value indicating all values of x are on the same chromosome. If x is a matrix, the values may be applied by row or by column. If by row, chr should be equal to dim(x)[1] and if by column, chr should be equal to dim(x)[2]. A single value may be used if all values are on the same chromosome.

The offset is calculated through the band.info object. See band.info for more information. If a bandobj is not specified the package's default bandinfo object is used.

Value

vector or matrix, of same form as input, of converted values

Author(s)

Lori A. Shepherd

See Also

Band.Info

Examples

```
#
# this is a mock example
#

x = matrix(0,24,3)
x[,2] = 1
x[,3] = 2

# this is x before
x

chr = 1:24

newx = convertGloc(x=x, chr=chr, row=TRUE, bandobj=NA)

# this is x after
newx
```

---

cytoBand *Band Information File*

Description

This file contains band information

Usage

```
data(cytoBand)
```

Format

a tab delimited text file with columns for chromosome, start location, stop location, and band.

Details

A band information file must minimally contain columns for chromosome, start and stop location, and band. see `bandinfo` and `band.info`

Note

see also `bandinfo`, `band.info`, `makeBandInfo`

References

Karolchik D, Kuhn, RM, Baertsch R, Barber GP, Clawson H, Diekhans M, Giardine B, Harte RA, Hinrichs AS, Hsu F, Miller W, Pedersen JS, Pohl A, Raney BJ, Rhead B, Rosenbloom KR, Smith KE, Stanke M, Thakkapallayil A, Trumbower H, Wang T, Zweig AS, Haussler D, Kent WJ. The UCSC Genome Browser Database: 2008 update. Nucleic Acids Res. 2008 Jan;36:D773-9

`DiseaseGenes` *Disease Gene Annotation File*

Description

This file is an annotation file for known disease genes. It has been altered slightly to include hyperlinks to the UCSC Genome Browser.

Usage

```
data(DiseaseGenes)
```

Format

tab delimited text file with columns for gene name, chromosome, start location, end location, and weblink to UCSC Genome Browser.

Details

annotation files must minimally have columns for gene, chromosome, start location, and end location. This file was altered to also include weblinks to the human genome browser at UCSC. see also `annotation`, `anninfo`, and `annObj`

Note

see also `annotation`, `anninfo`, and `annObj`

References

Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. Hamosh A, Scott AF, Amberger J, Bocchini C, Valle D, McKusick VA. Nucleic Acids Res. 2002 Jan 1;30(1):52-5

Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D. The human genome browser at UCSC. Genome Res. 2002 Jun;12(6):996-1006

DNArepairgenes *Cancer Gene Annotation File*

Description

This file is an annotation file for known DNA repair genes. It has been altered slightly to include hyperlinks to the UCSC Genome Browser.

Usage

```
data(DNArepairgenes)
```

Format

tab delimited text file with columns for gene name, chromosome, start location, end location, and weblink to UCSC Genome Browser.

Details

annotation files must minimally have columns for gene, chromosome, start location, and end location. This file was altered to also include weblinks to the human genome browser at UCSC. see also `annotation`, `anninfo`, and `annObj`

Note

see also `annotation`, `anninfo`, and `annObj`

References

Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D. The human genome browser at UCSC. Genome Res. 2002 Jun;12(6):996-1006

GGVobj-class *GGV 'generic genomic viewer' class*

Description

The generic genomic viewer object contains all necessary information to plot a series of interactive genomic plots.

GGVobj structure

values Contains values, objects, and matricies with main data source

interactive contains matricies and vectors of interactive data for the plots

info contains values of other arguments used in functions call

GGVobj's values structure

vls a matrix of values for the heatmap. assumes full set do not subset based on plot.x.index or plot.y.index, the function will automatically subset. The number of rows should correspond to the number of spot IDs in the mapObj

mapObj an object of the class mapobj

annObj an annotation object

chrArms Character vector of chromosome arms to be plotted. These names should match arm names in the mapObj's band.info. These Chromosomes will be diplayed in an index file

trackRegions A list containing character vectors for Broad.Band, Fine.Band, and geneName. Also may contain a n by 2 matrix of genomic start and stop values. These are known regions of interested that will be displayed as tiledImages. See makeTrack

mat matrix indicating layout. This argument will be passed into the graphics package layout call as mat.Each value in the matrix must be '0' or a positive integer. If N is the largest positive integer in the matrix, then the integers 1,...,N-1 must also appear at least once in the matrix. '0' indicates region of no plotting. This may be left as NA, and a default will be used. This matrix will be used for Chromosome Arm and Sub.Arm Plots

plot.call character vector containing extra plotting calls that will be evaluated on the optional side plot. This plot is added to the right of the annoation tracks. It is designed to add additional statistical analysis such as p.values, linear order statistics, etc.

plot.vec vector of values that will be plotted in the side plot. If NA, no plot will be added to the display. These values are the additional plots x-values. See details

plot.dx Index to subset plot.vec on when creating an initial genomic plot. If NA, all values are used

GGVobj's interactive structure

x.labels data frame of n x m which contains values relating to the x-axis values. This information is displayed in the interactive plot window

y.labels data frame of n x m which contains values relating to the y-axis values. This information is displayed in the interactive plot window

xy.labels list of matricies. All matricies should be of n x m. This information is displayed in the interactive plot window

x.links data frame of n x m which contains web addresses for links relating to the x-axis values. This information is displayed as hyperlinks in the interactive plot window

y.links data frame of n x m which contains web addresses for links relating to the y-axis values. This information is displayed as hyperlinks in the interactive plot window

xy.links list of matricies. All matricies should be of n x m. This information is displayed in the interactive plot window as hyperlinks. The values in these matricies should be complete web address

asLinks contains complete web address for points that should be treated as hyperlinks

x.images data frame of n x m which contains paths to images relating to the x-axis values. This information is displayed as images in the interactive plot window

y.images data frame of n x m which contains paths to images relating to the y-axis values. This information is displayed as images in the interactive plot window

xy.images list of matricies. All matricies should be of n x m. This information is displayed in the interactive plot window as images. The values in these matricies should be complete paths to images.

GGVobj's info structure

maxLabels maximum number of labels to appear on the heatmap y axis

mai.mat n x 4 matrix of values to be passed in for each plots par mai. n will be 3 if plot.call is NA, and 4 if plot.calls is specified. This will be used for Chromosome Arm and Sub.Arm plots

mai.prc logical indicating if mai mat values are percentages or hard coded values. If mai.prc is T, indicates percentage. This will be used for Chromosome Arm and Sub.Arm plots

plot.extras List of length equal to the number of plots: 3 if plot.call is NA, 4 if plot.call is specified. This object is a list of lists. The sublists contain any additional plotting calls that should be executed for the plot. Each entry must be a character vector. If no additional plotting is equired, an NA should be used

smpLines logical indicating if vertical lines should be added between each sample of the heatmap

divCol If smpLines, the color of the dividing lines

lims Lower and Upper limit for vls

annotation Numeric indication of which annotation information objects to include from the annObj. If NA all are used

clrs Character vector of colors to use for annotation track

mapObj.columns Which columns from mapObj's mapping.info data.frame to include in tool-tip. May be numeric or header names

Note

Users are encouraged to view the vignette for more detail

Author(s)

Lori A. Shepherd

See Also

`makeGGV`, `initGGV`, `updateGGV`

HB19Kv2.HG18 *Human Genome 19K BAC build Mapping File*

Description

This file is a mapping file for the human genome 19K BAC build.

Usage

```
data(HB19Kv2.HG18)
```

Format

tab delimited text file with columns for BAC name, chromosome, start location, stop location, central location, genomic location, band, mapped by, flag, and weblink to UCSC Genome Browser.

Details

mapping files must minimally contain columns for name, chromosome, and location. see also `mappingObj`, `mapping.info`, and `mapobj`

Note

see also `mappingObj`, `mapping.info`, and `mapobj`

References

The March 2006 human reference sequence (NCBI Build 36.1, Hg18). http://www.ncbi.nlm.nih.gov/genome/guide/human/release_notes.html#b36

Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D. The human genome browser at UCSC. Genome Res. 2002 Jun;12(6):996-1006

`iGGV` *Interactive Genomic Heatmap Display*

Description

This function creates an interactive genomic heatmap with any number of additional annotation tracks.

Usage

```
iGGV(vls,
      mapObj,
      annObj,
      x.labels=NA,
      y.labels=NA,
      xy.labels=NA,
      x.links=NA,
      y.links=NA,
      xy.links=NA,
      asLinks=NA,
      x.images=NA,
      y.images=NA,
      xy.images=NA,
      mat=NA,
      maxLabels=25,
      mai.mat = NA,
```

```
mai.prc=FALSE,
plot.x.index=NA,
smp.color = NA,
plot.y.index=NA,
goodDX=NA,
genomic.start=NA,
genomic.stop=NA,
genomic.region=NA,
region.type="chrom",
plot.extras=NA,
smpLines=TRUE,
divCol="lightgrey",
plot.call=NA,
plot.vec=NA,
lims = c(-0.5,0.5),
annotation = NA,
clrs=c("blue", "hotpink", "purple", "orange"),
mapObj.columns = NA,
fname.root="iGGV",
dir="./",
overwriteSourcePlot = NA,
makeInteractive=TRUE,
overrideInteractive=NA,
header="v3",
window.size = "800x1100",
image.size= "800x1100",
ps.paper="letter",
ps.width=8,
ps.height=11,
cleanDir=TRUE,
...)
```

Arguments

vls	a matrix of values for the heatmap. assumes full set do not subset based on plot.x.index or plot.y.index, the function will automatically subset. The number of rows should correspond to the number of spot IDs in the mapObj.
mapObj	an object of the class mapobj
annObj	an annotation object
x.labels	data frame of n x m which contains values relating to the x-axis values. This information is displayed in the interactive plot window.
y.labels	data frame of n x m which contains values relating to the y-axis values. This information is displayed in the interactive plot window.
xy.labels	list of matricies. All matricies should be of n x m. This information is displayed in the interactive plot window.
x.links	data frame of n x m which contains web addresses for links relating to the x-axis values. This information is displayed as hyperlinks in the interactive plot window.

y.links	data frame of n x m which contains web addresses for links relating to the y-axis values. This information is displayed as hyperlinks in the interactive plot window.
xy.links	list of matricies. All matricies should be of n x m. This information is displayed in the interactive plot window as hyperlinks. The values in these matricies should be complete web address.
asLinks	contains complete web address for points that should be treated as hyperlinks.
x.images	data frame of n x m which contains paths to images relating to the x-axis values. This information is displayed as images in the interactive plot window.
y.images	data frame of n x m which contains paths to images relating to the y-axis values. This information is displayed as images in the interactive plot window.
xy.images	list of matricies. All matricies should be of n x m. This information is displayed in the interactive plot window as images. The values in these matricies should be complete paths to images.
mat	matrix indicating layout. This argument will be passed into the graphics package layout call as mat.Each value in the matrix must be '0' or a positive integer. If N is the largest positive integer in the matrix, then the integers {1,...,N-1} must also appear at least once in the matrix. '0' indicates region of no plotting. This may be left as NA, and a default will be used
maxLabels	maximum number of labels to appear on the heatmap y axis
mai.mat	n x 4 matrix of values to be passed in for each plots par mai. n will be 3 if plot.call is NA, and 4 if plot.calls is specified
mai.prc	logical indicating if mai mat values are percentages or hard coded values. If mai.proc is T, indicates percentage.
plot.x.index	subset of x values to use
smp.color	color for x-axis header values
plot.y.index	subset of y values to use
goodDX	range of acceptable y values. This is used as a way to removed spots of known bad quality. should correspond to locations in the mapObj\$mapping.info object
genomic.start	numeric value indicating starting genomic location for region to plot. Note: will round to nearest fine.band region
genomic.stop	numeric value indicating ending genomic location for region to plot. Note: will round to nearest fine.band region
genomic.region	character value of desired region to plot. Given in chromosome, arm, broad.band, or fine.band. i.e 1, 2p, 1p13.1, etc.
region.type	If genomic region is used, the type of region given. May either be, chrom, arm, broad.band, or fine.band.
plot.extras	List of length equal to the number of plots: 3 if plot.call is NA, 4 if plot.call is specified. This object is a list of lists. The sublists contain any additional plotting calls that should be executed for the plot. Each ntry must be a character vector. If no additional plotting is equired, an NA should be used
smpLines	logical indicating if vertical, yellow lines should be added between each sample of the heatmap.

divCol	If smpLines, the color of the dividing lines
plot.call	character vector containing plot call that will be evaluated. This plot is added to the right of the annoation tracks. It is designed to add additional statistical analysis such as p.values, linear order statistics, etc. If NA, no plot will be added to the display
plot.vec	vector of values that will be plotted in plot.call. While this may be specified in the plot.call, it is also necessary in order to add interactive tool-tip to values
lims	Lower and Upper limit for vls
annotation	Numeric indication of which annotation information objects to include from the annObj. If NA all are used.
clrs	Character vector of colors to use for annotation tracks
mapObj.columns	Which columns from mapObj's mapping.info data.frame to include in tool-tip. May be numeric or header names
fname.root	Base name to use for all file created
dir	directory path to where files should be created
overwriteSourcePlot	character, should static image generated be a postscript, png, jpeg, or tiff. see makeSplot for more details
makeInteractive	logical, should interactive html file be created.see makeSplot for more details
overrideInteractive	Indicates which figures of the layout should be interactive. see makeSplot for more details
header	May either be v1,v2, or v3. Determines which tooltip header will be in the html file. see makeSplot for more details
window.size	size of the html window. see makeSplot for more details
image.size	character indicating resize value of image,'width'x'height' see initSplot for more details
ps.paper	character indicating paper for postscript function
ps.width	width argument for postscript function
ps.height	hieght argument for postscript function
cleanDir	logical indicating if intermediate files created for mapping purposes should be deleted
...	extra arguments to makeImap function

Details

This function is a wrapper to sendplot to create an interactive bioinformatics heatmap display.

The function requires a matrix of values for the heatmap, a mapping object, and an annotation object. For details on creating the mapping and annoation objects see mappingObj and annotation.

The matrix of values should be a complete listing, do not subset based on plot.x.index or plot.y.index. The function will subset automatically.

The matricies x.labels, y.labels, xy.labels, x.links, y.links, xy.links,asLinks, x.images, y.images, and xy.images should also be complete listings. Do not subset based on plot.x.index or plot.y.index. The function will subset automatically.

There are three ways to indicate y values that should be plotted. They may be specified directly through the plot.y.index. This is a numeric vector which corresponds to the ordering in the mapping object. They may be determined through giving a genomic starting and ending location, genomic.start and genomic stop respectively. Both starting and ending locations must be given if this option is utilized. The genomic locations should be the genomic location with respect to the entire genome not within a chromosome. Finally, they may be specified by listing a single specific region to be plotted with genomic.region. If this option is used, the user must also indicate what type of region is listed in the region.type argument. The four options for this argument are chrom, arm, broad.band, fine.band. The region given should match up to a region in the mapping object.

It is possible to control what is displayed from the annotation object so that any, all, or none of the annotation information may be displayed as tracks along side the heatmap. The annotation argument is a numeric corresponding to the order of the annotation information objects in the annObj. NA will display all.

Value

.png [.jpeg,.ps] and .html graphs. .html is an interactive display. The display will have a heatmap, legend, annotation track, and, if utilized, an extra descriptive plot.

Note

Works properly on linux/unix operating systems only. The below example show one way to use iGGV as far as plotting and subsetting. Please see vignette for other examples.

Author(s)

Lori A. Shepherd

See Also

sendplot, annotation, mappingObj, iGGVex

Examples

```
library("iGenomicViewer")

data(mapping.info)
data(annObj)
data(iGGVex)


iGGV(vls = mat,
     mapObj=mapping.info,
     annObj=annObj,
     x.labels=x.lbls,
     y.labels=y.lbls,
```

```
genomic.region="11q",
region.type="arm",
mapObj.columns =c(1,2,3,4))
```

---

iGGVex *Example Dataset for iGGV Function*

Description

This dataset contains an example data matrix of values that corresponds to the default mapping object. It also contains mock description data for the spot.ID and for samples, y.lbls and x.lbls respectively.

Usage

```
data(iGGVex)
```

Format

mat is a matrix of values. The number of rows should correspond to the number of spots in the mapping object. It this case the default mapping object is used. see `mapping.info`

y.lbls is a matrix with descriptive attributes for the spots. The number of rows should correspond to the number of spots in the mapping object. It this case the default mapping object is used. see `mapping.info`. Do not subset based on the desired range; the program assumes a complete table and will automatically subset.

x.lbls is a matrix with descriptive attributes for the samples. The number of rows should be equal to the number of samples, or the number of columns in mat. Do not subset based on the desired range; the program assumes a complete table and will automatically subset.

Note

mat matrix corresponds to default mapping object see `mapping.info`. mock data

---

iGGVtiled *Interactive Tiled Display*

Description

This function takes a tiled image display 'TIplot' object and creates an interactive layout of plots. The tiled display will be the main plot of the layout.

Usage

```
iGGVtiled(TIplot,
          annObj,
          x.labels=NA,
          y.labels=NA,
          xy.labels=NA,
          x.links=NA,
          y.links=NA,
          xy.links=NA,
          asLinks=NA,
          x.images=NA,
          y.images=NA,
          xy.images=NA,
          mat=NA,
          mai.mat = NA,
          mai.prc=FALSE,
          plot.extras=NA,
          smpLines=TRUE,
          divCol="lightgrey",
          plot.call=NA,
          plot.vec=NA,
          lims = c(-0.5,0.5),
          annotation = NA,
          clrs=c("blue", "hotpink", "purple", "orange"),
          mapObj.columns = NA,
          fname.root="iGGV",
          dir="./",
          overwriteSourcePlot = NA,
          makeInteractive=TRUE,
          overrideInteractive=NA,
          header="v3",
          window.size = "800x1100",
          image.size= "800x1100",
          cleanDir=TRUE,
          vrb=TRUE,
          ...)
```

Arguments

TIplot	an object of the class TIplot. This object's genomic locations must be with respect to the entire genome not location within chromosome.
annObj	an annotation object
x.labels	data frame of n x m which contains values relating to the x-axis values. This information is displayed in the interactive plot window.
y.labels	data frame of n x m which contains values relating to the y-axis values. This information is displayed in the interactive plot window.

xy.labels	list of matricies. All matricies should be of n x m. This information is displayed in the interactive plot window.
x.links	data frame of n x m which contains web addresses for links relating to the x-axis values. This information is displayed as hyperlinks in the interactive plot window.
y.links	data frame of n x m which contains web addresses for links relating to the y-axis values. This information is displayed as hyperlinks in the interactive plot window.
xy.links	list of matricies. All matricies should be of n x m. This information is displayed in the interactive plot window as hyperlinks. The values in these matricies should be complete web address.
asLinks	contains complete web address for points that should be treated as hyperlinks.
x.images	data frame of n x m which contains paths to images relating to the x-axis values. This information is displayed as images in the interactive plot window.
y.images	data frame of n x m which contains paths to images relating to the y-axis values. This information is displayed as images in the interactive plot window.
xy.images	list of matricies. All matricies should be of n x m. This information is displayed in the interactive plot window as images. The values in these matricies should be complete paths to images.
mat	matrix indicating layout. This argument will be passed into the graphics package layout call as mat.Each value in the matrix must be '0' or a positive integer. If N is the largest positive integer in the matrix, then the integers {1,...,N-1} must also appear at least once in the matrix. '0' indicates region of no plotting. This may be left as NA, and a default will be used
mai.mat	n x 4 matrix of values to be passed in for each plots par mai. n will be 3 if plot.call is NA, and 4 if plot.calls is specified
mai.prc	logical indicating if mai mat values are percentages or hard coded values. If mai.proc is T, indicates percentage.
plot.extras	List of length equal to the number of plots: 3 if plot.call is NA, 4 if plot.call is specified. This object is a list of lists. The sublists contain any additional plotting calls that should be executed for the plot. Each ntry must be a character vector. If no additional plotting is equired, an NA should be used
smpLines	logical indicating if vertical, yellow lines should be added between each sample of the heatmap.
divCol	If smpLines, the color of the dividing lines
plot.call	character vector containing plot call that will be evaluated. This plot is added to the right of the annoation tracks. It is designed to add additional statistical analysis such as p.values, linear order statistics, etc. If NA, no plot will be added to the display
plot.vec	vector of values that will be plotted in plot.call. While this may be specified in the plot.call, it is also necessary in order to add interactive tool-tip to values
lims	Lower and Upper limit for vls
annotation	Numeric indication of which annotation information objects to include from the annObj. If NA all are used.

clrs	Character vector of colors to use for annotation tracks
mapObj.columns	Which columns from mapObj's mapping.info data.frame to include in tool-tip. May be numeric or header names
fname.root	Base name to use for all file created
dir	directory path to where files should be created
overwriteSourcePlot	character, should static image generated be a postscript, png,jpeg, or tiff. see makeSplot for more details
makeInteractive	logical, should interactive html file be created.see makeSplot for more details
overrideInteractive	Indicates which figures of the layout should be interactive. see makeSplot for more details
header	May either be v1,v2, or v3. Determines which tooltip header will be in the html file. see makeSplot for more details
window.size	size of the html window. see makeSplot for more details
image.size	character indicating resize value of image,'width'x'height' see initSplot for more details
cleanDir	logical indicating if intermediate files created for mapping purposes should be deleted
vrb	logical indicating if status messages should be printed
...	extra arguments to makeImap function

Details

This function is a wrapper to sendplot to create an interactive bioinformatics heatmap display.

The main plot of the layout is a tiled display image. The reasoning behind the tiled image is that there are spot.ID, BAC clones, etc. that span the genome. The different regions may overlap or not include certain regions across the entire genome. This viewer breaks the spot.IDs to show how they are covering a particular region.

This function requires an object of the class TIplot, a mapping object, and an annotation object. For details on creating the mapping and annoation objects see mappingObj and annotation. For details on creating the TIplot object see initTile. The TIplot start and stop genomic locations must be with respect to the entire genome not with chromosome. If the TIplot object was created with the later, plese see convertLoc for help on converting these locations and recreate the object.

It is possible to control what is displayed from the annotation object so that any, all, or none of the annotation information may be displayed as tracks along side the heatmap. The annotation argument is a numeric corresponding to the order of the annotation information objects in the annObj. NA will display all. 0 will display none.

For more details on interactive plotting capabilities, please refer to vignette.

Value

.png [.jpeg,.ps] and .html graphs. .html is an interactive display. The display will have a tiled heatmap, legend, annotation track, and, if utilized, an extra descriptive plot.

Note

Works properly on linux/unix operating systems only.

The TIplot object must have Y locations with respect to entire genome not within chromosome. See `convertLoc` for conversion option.

Author(s)

Lori A. Shepherd, Daniel P. Gaile

See Also

`initTile`,`makeTiled`, `iGGV`

Examples

```
library("iGenomicViewer")

# load data objects
data(iGGVex)
data(mapping.info)
mapObj = mapping.info

# create a subset range
bacDX = 103:112
smplDX = 1:10
Z = mat[bacDX,smplDX]


# make object
TIplot = initTile(Z=Z,
                  bacDX=bacDX, mapObj=mapObj,
                  H=3,zlims=c(-0.5,0.5),
                  ylabels=paste("Spot",bacDX, sep=""),
                  xlabels=paste("smp",smplDX, sep=""),
                  xlab="Samples",
                  ylab="SpotID",
                  ttl="tiledImage",
                  returnVl=TRUE,
                  saveFlag=FALSE,
                  saveName="TIplot.RData")


# annotaiton object
# see vignette and help files for more details
data(annObj)

# make interactive plot

iGGVtiled(TIplot=TIplot,
          annObj=annObj,
```

```
x.labels=as.data.frame(list(
        sample.ID=paste("smp",1:TIplot$vls$nsmp,sep=""),
        xla1=c("a","b","c","d","e","f","g","h","i","j"),
        xla2=10:1)),
y.labels=as.data.frame(list(
        Spot.ID=paste("Spot",bacDX,sep=""))),
xy.labels=list(lgr=round(Z,3)),
mapObj.columns = c(2,3,7),
fname.root="iGGVtiled")
```

`initGGV` *Initialize GGV Object*

Description

This function initializes a GGV object to use with makeGGV

Usage

```
initGGV(vls,
        mapObj,
        annObj,
        x.labels=NA,
        y.labels=NA,
        xy.labels=NA,
        x.links=NA,
        y.links=NA,
        xy.links=NA,
        asLinks=NA,
        x.images=NA,
        y.images=NA,
        xy.images=NA,
        chrArms=NA,
        trackRegions=NA,
        side.plot.extras=NA,
        plot.vec=NA,
        plot.dx=NA,
        maxLabels=25,
        mat = NA,
        mai.mat = NA,
        mai.prc=FALSE,
```

```
plot.extras=NA,
smpLines=TRUE,
divCol="lightgrey",
lims = c(-0.5,0.5),
annotation = NA,
clrs=c("blue", "hotpink", "purple", "orange"),
mapObj.columns = NA,
returnVl=TRUE,
saveFlag=FALSE,
saveName="GGVobj.RData")
```

Arguments

vls	a matrix of values for the heatmap. assumes full set do not subset based on a desired sample or spot index, opportunities to subset will be given in executing the plots and the function will automatically subset. The number of rows should correspond to the number of spot IDs in the mapObj
mapObj	an object of the class mapobj
annObj	an annotation object
x.labels	data frame of n x m which contains values relating to the x-axis values. This information is displayed in the interactive plot window
y.labels	data frame of n x m which contains values relating to the y-axis values. This information is displayed in the interactive plot window
xy.labels	list of matricies. All matricies should be of n x m. This information is displayed in the interactive plot window
x.links	data frame of n x m which contains web addresses for links relating to the x-axis values. This information is displayed as hyperlinks in the interactive plot window
y.links	data frame of n x m which contains web addresses for links relating to the y-axis values. This information is displayed as hyperlinks in the interactive plot window
xy.links	list of matricies. All matricies should be of n x m. This information is displayed in the interactive plot window as hyperlinks. The values in these matricies should be complete web address
asLinks	contains complete web address for points that should be treated as hyperlinks
x.images	data frame of n x m which contains paths to images relating to the x-axis values. This information is displayed as images in the interactive plot window.
y.images	data frame of n x m which contains paths to images relating to the y-axis values. This information is displayed as images in the interactive plot window.
xy.images	list of matricies. All matricies should be of n x m. This information is displayed in the interactive plot window as images. The values in these matricies should be complete paths to images.
chrArms	Character vector of chromosome arms to be plotted. These names should match arm names in the mapObj's band.info. These Chromosomes will be diplayed in an index file

trackRegions	A list containing character vectors for Broad.Band, Fine.Band, and geneName. Also may contain a n by 2 matrix of genomic start and stop values. These are known regions of interested that will be displayed as tiledImages. See makeTrack
side.plot.extras	character vector containing extra plotting calls that will be evaluated on the optional side plot. This plot is added to the right of the annoation tracks. It is designed to add additional statistical analysis such as p.values, linear order statistics, etc.
plot.vec	vector of values that will be plotted in the side plot. If NA, no plot will be added to the display. These values are the additional plots x-values. See details
plot.dx	Index to subset plot.vec on when creating an initial genomic plot. If NA, all values are used
maxLabels	maximum number of labels to appear on the y axis
mat	matrix indicating layout. This argument will be passed into the graphics package layout call as mat.Each value in the matrix must be '0' or a positive integer. If N is the largest positive integer in the matrix, then the integers {1,...,N-1} must also appear at least once in the matrix. '0' indicates region of no plotting. This may be left as NA, and a default will be used. This matrix will be used for Chromosome Arm and Sub.Arm Plots
mai.mat	n x 4 matrix of values to be passed in for each plots par mai. n will be 3 if plot.vec is NA, and 4 if plot.vec is specified. This will be used for Chromosome Arm and Sub.Arm plots
mai.prc	logical indicating if mai mat values are percentages or hard coded values. If mai.prc is T, indicates percentage. This will be used for Chromosome Arm and Sub.Arm plots
plot.extras	List of length equal to the number of plots: 3 if plot.vec is NA, 4 if plot.vec is specified. This object is a list of lists. The sublists contain any additional plotting calls that should be executed for the plot. Each entry must be a character vector. If no additional plotting is equired, an NA should be used
smpLines	logical indicating if vertical lines should be added between each sample of the heatmap
divCol	If smpLines, the color of the dividing lines
lims	Lower and Upper limit for vls
annotation	Numeric indication of which annotation information objects to include from the annObj. If NA all are used
clrs	Character vector of colors to use for annotation tracks
mapObj.columns	Which columns from mapObj's mapping.info data.frame to include in tool-tip. May be numeric or header names
returnVl	Should GGV object be returned
saveFlag	Should GGV object be saved
saveName	If saveFlag, path file name to save object

Details

trackRegions list known regions of interest that should show more descriptive plots. Regions may be specified by Broad.Band, Fine.Band, genomic location, and geneName. See makeTrack for more details

plot.vec will be subset based on plot.dx to make the genomic plot in the makeGGV function. It is important the the plot.vec are across the genome so the additional plot may be displayed for all chromosome arms. The vector contains x-values to plot. The y-values will be determined by plot.dx, or internal subsetting. If plot.vec is NA no side plot will be included

side.plot.extras is a character string. It is similar to the plot.extras argument only it is a single variable. These calls will be evaluated for the additional side plot, if utilized. The calls should be separated by a semicolon.

See vignette for more details

Value

If returnVl, an object of the class 'GGVobj', a Genomic Viewer Object, is returned

Note

see vignette for more details

Author(s)

Lori A. Shepherd

See Also

GGVobj, iGGV, makeTrack

Examples

```
# load library
library("iGenomicViewer")

# load example data
data(iGGVex)
data(mapping.info)
data(annObj)


mapObj = mapping.info
y.lbls$Pdisc = round(y.lbls$Pdisc,3)
chrArm = c("8p", "18p")
pvls = rep(rep(rep(1:4, each=5), 960))[1:length(mapObj$mapping.info$g.loc.center)]
plot.vec = pvls[1:length(mapObj$mapping.info$g.loc.center)]
side.plot.extras="title(main='pvals')"


# make trackregion object
```

```
trackRegion = makeTrack(Fine.Band = c("8p11.22","18p11.21"),genomicLoc = NA, geneName = "FANCE")

# initialize GGV object
GGV = initGGV(vls = round(mat, 3),
              mapObj = mapObj,
              annObj = annObj,
              x.labels=x.lbls,
              y.labels=y.lbls,
              xy.labels=NA,
              chrArms=chrArm,
              trackRegions=trackRegion,
              side.plot.extras=side.plot.extras,
              plot.vec=plot.vec,
              plot.dx=which(mapObj$mapping.info$Chrom=="chr8"),
              mapObj.columns =c(2,8,10),
              smpLines=TRUE,
              divCol="lightgrey")
```

`initTile` *Creates A Tiled Display Image 'TIplot' Object*

Description

The initTile function creates a tiled display image 'TIplot' object. A TIplot object holds all necessary elements to make a tiled display plot.

Usage

```
initTile(Z,
         bacDX,
         goodDX=NA,
         mapObj=NA,
         H=2,
         zlims=c(-0.5,0.5),
         smplDX=NA,
         ylabels=NA,
         xlabels=NA,
         x.axis.cex =0.5,
         y.axis.cex =0.5,
         xlab="samples",
         ylab="BAC location",
         ttl=NA,
         returnVl=TRUE,
         saveFlag=FALSE,
         saveName="TIplot.RData")
```

Arguments

Z	matrix of values for image. The number of rows and columns should be equal to the lenghts of bacDX and smplDX. If the matrix is larger the matrix will be subset based on bacDX and smplDX
bacDX	index of spot.ID's to graph. should correspond to index in mapObj. This will be used to determine the genomic start and stopping location of the plot
goodDX	range of acceptable y values. This is used as a way to removed spots of known bad quality. should correspond to locations in the mapObj\$mapping.info object
mapObj	an object of the class mapobj. This object should be made with genomic starting and stopping locations not a central location
H	number of tracks to display per sample
zlims	minimum and maximum range for values in Z
smplDX	subset for x axis. If the second dimension of Z is larger than or equal to the length of smplDX, Z will be subset based on this index
ylabels	vector indicating labels for Y axis. Should be equal in length to the number of rows in Z [or Y]
xlabels	vector indicating labels for X axis. Should be equal in length to the number of columns in Z
x.axis.cex	display size of xlabels
y.axis.cex	display size of ylabels
xlab	main x axis label for plot
ylab	main y axis label for plot
ttl	main title for plot
returnVl	Should TIplot object be returned
saveFlag	Should TIplot object be saved
saveName	If saveFlag, path file name to save object

Details

The initTile function creates a tiled display image 'TIplot' object. A TIplot object holds all necessary elements to make a tiled display plot. The function takes in a matrix of values to display as a tiled image, Z,and a bacDX indicating the spot.IDs that should be used and correspond to the rows of Z. If Z has more rows than the length of bacDX, it is assumed that Z should be subset based on bacDX. The bacDX should refer to spot.IDs given in a mapObj. If no mapObj is used, the default package mapObj is used. It is also possible to subset Z based on a smplDX. If no smplDX is given it is assumed that all columns of Z should be used. The reasoning behind the tiled image is that there are spot.ID, BAC clones, etc. that span the genome. The different regions may overlap or not include certain regions across the entire genome. This viewer breaks the spot.IDs to show how they are covering a particular region.

Value

If returnVl, an object of the class 'TIplot'

Note

Assumes genomic location values on Y axis and samples on X axis

Author(s)

Lori A. Shepherd, Daniel P. Gaile

See Also

convertLoc, TIplot, makeTile, iGGVtiled

Examples

```
library("iGenomicViewer")

# load data objects
data(iGGVex)
data(mapping.info)
mapObj = mapping.info

# create a subset range
bacDX = 103:112
smplDX = 1:10
Z = mat[bacDX,smplDX]

# make object
TIplot = initTile(Z=Z,
                  bacDX=bacDX, mapObj=mapObj,
                  H=3,zlims=c(-0.5,0.5),
                  ylabels=paste("Spot",bacDX, sep=""),
                  xlabels=paste("smp",smplDX, sep=""),
                  xlab="Samples",
                  ylab="SpotID",
                  ttl="tiledImage",
                  returnVl=TRUE,
                  saveFlag=FALSE,
                  saveName="TIplot.RData")
```

---

makeBandInfo *Function for creating bandinfo object*

Description

This function creates a bandinfo object that contains genomic location information for chromosomes, arms, broad bands, and fine bands.

Usage

```
makeBandInfo(file,
             chrom.levels,
             file.sep="\t",
             autosomes=1:22,
             X.chrom = 23,
             Y.chrom = 24,
             chr.dx = 1,
             band.dx = 4,
             start.dx = 2,
             stop.dx = 3,
             returnVl=TRUE,
             saveFile=FALSE,
             saveName = "BandInfo.RData",
             ...)
```

Arguments

file	path name of file
chrom.levels	vector indicating how the chrom column in the file is represented. i.e chr1, chrom1, 1, etc.
file.sep	seperation character for file
autosomes	numeric vector listing autosomes. defaults to human 1:22
X.chrom	numeric indication of x chromosome. defaults to human 23
Y.chrom	numeric indication of y chromosome. defaults to human 24
chr.dx	column in file that represents the chromosome
band.dx	column in file that represents the band. assumed to be without chromosome i.e. q11, p13.1, etc.
start.dx	column in file that represents the starting genomic location of band region
stop.dx	column in file that represents the stopping genomic location of band region
returnVl	logical indicating if object created should be returned
saveFile	logical indicating if object created should be saved to a file
saveName	if saveVl, path name of file
...	additional arguments to the read.table function for reading the file

Details

This function is designed to make useful data.frames of starting and ending genomic locations for chromosomes, arms, broad.bands, and fine.bands. Each data.frame has five columns: the region (chrom, arm), label, lower, center, and upper. lower, center, and upper refer to genomic locations. label is the label that will be used for the region when plotting. The bandinfo object will also contain an offset object. This is a numeric vector of length equal to the number of chromsomes. It contains the numeric buffer that should be added to genomic locations given within chromosome to get the location with respect to the entire genome. These data.frames and indicies aid in graphing when utilizing the iGGV function.

Value

a bandinfo object that contains location information for chormosomes, arms, broad bands, and fine bands. see bandinfo for more details

Author(s)

Lori A. Shepherd

See Also

bandinfo, Band.Info, iGGV

Examples

```
#
# The following code will build the same object as data(Band.Info)
#

library("iGenomicViewer")

# writes out text files to use
writeExFiles()

# make object
band.info = makeBandInfo(file="cytoBand.txt", chrom.levels=c("chr1","chr2","chr3","chr4","chr5","chr6","chr7","
```

---

makeGGV *Creates a Series of Interactive Plots*

Description

This functions acts on a GGV 'generic genomic viewer' object to create a series of interactive plots.

Usage

```
makeGGV(GGV,
        goodDX=NA,
        smplDX=NA,
        smp.color=NA,
        break.num=125,
        tileNum = 2,
        buffer = 5,
        makeWinArms=TRUE,
        tiledMat=NA,
```

```
        tiledMai.mat = NA,
        tiledMai.prc=FALSE,
        fname.root="iGGV",
        dir="GGV/",
        overwriteSourcePlot = NA,
        header="v3",
        window.size = "800x1100",
        image.size = "800x1100",
        tiled.window.size = "800x1100",
        tiled.image.size = "1200x1100",
        cleanDir = TRUE)
```

Arguments

GGV	a GGVobj, 'generic genomic viewer' object
goodDX	range of acceptable y values. This is used as a way to removed spots of known bad quality. should correspond to locations in the mapObj\$mapping.info object
smplDX	index or ordering for x axis
smp.color	colors for x axis
break.num	numeric indicating break lengths to break up heatmap for interactivity
tileNum	For tiled image heatmaps, the number of tracks to split spot.IDs
buffer	An additional number of spots to plot surrounding know regions.
makeWinArms	Should within Arm sub plots be generated. This creates all chromosome sub plots for chromosomes listed
tiledMat	matrix indicating layout. This argument will be passed into the graphics package layout call as mat.Each value in the matrix must be '0' or a positive integer. If N is the largest positive integer in the matrix, then the integers {1,...,N-1} must also appear at least once in the matrix. '0' indicates region of no plotting. This may be left as NA, and a default will be used. This matrix will be used for tiled images only. mat for chromosome arms and sub arms are specified in initGGV
tiledMai.mat	n x 4 matrix of values to be passed in for each plots par mai. n will be 3 if plot.call is NA, and 4 if plot.calls is specified. This will be used for tiled images only. mai.mat for chromosome arms and sub arms are specified in initGGV
tiledMai.prc	logical indicating if mai mat values are percentages or hard coded values. If mai.prc is T, indicates percentage. This will be used for tiled images only. mai.prc for chromosome arms and sub arms are specified in initGGV
fname.root	Base name to use for index file and genomic plot if applicable.
dir	directory path to where files and sub directories should be created
overwriteSourcePlot	character, should static image generated be a postscript, png, jpeg, or tiff. see makeSplot for more details
header	May either be v1,v2, or v3. Determines which tooltip header will be in the html file. see makeSplot for more details
window.size	size of the html window for chromosome arm and sub chromosome arm plots. see makeSplot for more details

`image.size` character indicating resize value of image,'width'x'height' for chromosome arms and sub chromosome arm plots. see `initSplot` for more details

`tiled.window.size` size of the html window for tiled image plots.see `makeSplot` for more details

`tiled.image.size` character indicating resize value of image,'width'x'height' tiled image plots. see `initSplot` for more details

`cleanDir` logical indicating if temporary files should be removed

Details

see vignette for better details

This function creates a series of plots based on user settings. The top layer is a chromosome arm index page with listed arms in chrArms. If a genomic plot is specified in the GGVobj\$values\$plot.call, a genomic plot is also generated for the given subregion (based on GGVobj\$values\$plot.dx, GGVobj\$values\$plot.vec). While the initial plot will be plotted over the given plot.dx, it is important that the plot.vec and plot.call arguments are across the genome so the plot may be included on any chromosome arm plots specified. The next set of plots are the chromosome arm plots with limit interactivity. These plots will be generated for all chromosome arms listed in chrArm or utilized in the genomic plot plot.dx. The main heatmap is not interactive in these plots, only the annotation tracks and, if applicable, the genomic plot are interactive. Additional annotation tracks for known regions of interest as well as sub chromosome arm regions are added. The known regions are tiled image displays; these regions were specified in GGV\$values\$trackRegions. see makeTrack. If the user clicks on one of these regions the assocaited plot will appear. The sub chromosome arm regions are fully interactive plots. These plots are only generated if makeWinArms is TRUE. If makeWinArms is FALSE the links in these regions will not work properly.

Value

A series of interactive plots are generated within the given directory, dir

Note

see vignette for more details

Author(s)

Lori A. Shepherd

See Also

`GGVobj`,`initGGV`

Examples

```
# see vignette for example
```

makeTiled *Makes Tiled Image Display Plot*

Description

This function makes a tiled image display plot from a tiled image display 'TIplot' object.

Usage

```
makeTiled(TIplot,
          smpDiv=TRUE,
          divCol="lightgrey")
```

Arguments

TIplot	tiled display image object, of the class 'TIplot'
smpDiv	Should vertical dividing lines be placed in between samples of plot
divCol	If smpDiv, the color of the dividing line

Details

This function take the TIplot object and creates a static tiled display image. All information for plotting is stored in the TIplot object. The user may choose to display lines separating samples. This and the color of the separation lines are controlled by smpDiv and divCol.

Value

Generates a static tiled display plot

Author(s)

Lori A. Shepherd, Daniel P. Gaile

See Also

initTile, iGGVtiled

Examples

```
# load data objects
library("iGenomicViewer")
data(iGGVex)
data(mapping.info)

# create a subset range
bacDX = 103:112
smplDX = 1:10
```

```
Z = mat[bacDX,smplDX]

# make object
TIplot = initTile(Z=Z,
                  bacDX=bacDX, mapObj=mapping.info,
                  H=3,zlims=c(-0.5,0.5),
                  ylabels=paste("Spot",bacDX, sep=""),
                  xlabels=paste("smp",smplDX, sep=""),
                  xlab="Samples",
                  ylab="SpotID",
                  ttl="tiledImage",
                  returnVl=TRUE,
                  saveFlag=FALSE,
                  saveName="TIplot.RData")


# plot object
#      makeTiled(TIplot)
```

---

makeTrack *Creates a trackRegion object*

Description

This function creates a track region object. This object is utilized in makeGGV to store known genomic regions of interest.

Usage

```
makeTrack(Broad.Band=NA,
          Fine.Band=NA,
          genomicLoc=NA,
          geneName=NA)
```

Arguments

Broad.Band	character vector of broad band regions. These regions should match to an associated mapObj\$band.info\$Broad.Band
Fine.Band	character vector of fine band regions. These regions should match to an associated mapObj\$band.info\$Fine.Band
genomicLoc	numeric vector or matrix of genomic locations. see details
geneName	character vector of geneNames, or name in label column of an associated annObj's annotation data.frame.

Details

The makeTrack function makes a trackRegion object to store known regions of interest. There are four ways to speicify a region: broad band, fine band, genomic start and stop locations, and geneName.

Broad.Band and Fine.Band are character vectors containing names of broad band or fine band regions. They should match an entry in the broad.band or fine.band column in a mapObj\$band.info data frames object. The geneName character vector is associates with an annObj. This character vector should contain entries that match the labels in the columns of an annObj's annotation data.frame. The list may contain entries from any and all objects of the annObj. The genomicLoc entry may either be a numeric vector or matrix. If it is a vector it assumes the entries are listed in order of start loc, stop loc, start loc, stop loc, for given regions. If it is a matrix, it assumes the first column is start locations and the second is stop locations.

Any of the four entries may be left as NA.

A trackRegion object is returned. This object is a list of known regions, with entries for Broad.Band, Fine.Band, genomicLoc, and geneName.

Value

trackRegion object, list containing known regions of interest, see details

Note

see vignette for more details

Author(s)

Lori A. Shepherd

See Also

updateGGV, makeGGV

Examples

```
library("iGenomicViewer")

trackRegion = makeTrack(Fine.Band = c("8p11.22","18p11.21"),
                        genomicLoc = NA,
                        geneName = "FANCE")
```

mapobj-class *mapobj 'mapping object' class*

Description

The mapping object contains all mapping information which includes but is not limited to spotIDs, chromosome locations, and genomic locations. The mapping object will also have a band.info object associated with it in order to map spot.IDs to chromosome, arm, broad and fine bands.

mapobj object structure

band.info a band.info object. see `makeBandInfo`

mapping.info A data.frame of mapping information. The information included depends on the mapping file used and the user settins when creating the object with the mappingObj function. The data.frame must have Spot.ID, Chrom, as well as one of the options for genomic locations. see `mappingObj` for more details.

links a data frame of web address information for the mapping. It should have the same row dimensions as the mapping.info data.frame.

images a data frame of paths to images relating to the mapping. It should have the same row dimensions as the mapping.info data.frame.

Author(s)

Lori A. Shepherd

See Also

`mappingObj`

mapping.info *Default Object of Class mapobj*

Description

This dataset contains an object of the class mapobj.

Usage

```
data(mapping.info)
```

Format

see `mapobj` for more details

Note

see `mapobj` and `mappingObj`

mappingObj *Creates Mapping Object*

Description

This function creates a mapping object

Usage

```
mappingObj(file,
           spot.ID,
           chrom,
           chrom.levels,
           loc=NA,
           loc.start=NA,
           loc.stop=NA,
           file.sep="\t",
           additional=NA,
           names.additional=NA,
           links=NA,
           names.links=NA,
           images=NA,
           names.images=NA,
           band.info = NA,
           returnVl = TRUE,
           saveFile = FALSE,
           saveName="MapObj.RData",
           ...)

mappingObjMarray(obj,
                 spot.ID=NA,
                 chrom,
                 locBy,
                 base.chrm=NA,
                 reg.exp=NA,
                 loc=NA,
                 loc.start=NA,
                 loc.stop=NA,
                 additional=NA,
                 names.additional=NA,
                 links=NA,
                 names.links=NA,
                 images=NA,
                 names.images=NA,
                 band.info = NA,
                 returnVl = TRUE,
                 saveFile = FALSE,
```

```
                    saveName="MapObj.RData")

mappingObjADF(adf,
                spot.ID=NA,
                chrom,
                locBy,
                base.chrm=NA,
                reg.exp=NA,
                loc=NA,
                loc.start=NA,
                loc.stop=NA,
                additional=NA,
                names.additional=NA,
                links=NA,
                names.links=NA,
                images=NA,
                names.images=NA,
                band.info = NA,
                returnVl = TRUE,
                saveFile = FALSE,
                saveName="MapObj.RData")


mappingObjDF(df,
               spot.ID=NA,
               chrom,
               locBy,
               base.chrm=NA,
               reg.exp=NA,
               loc=NA,
               loc.start=NA,
               loc.stop=NA,
               additional=NA,
               names.additional=NA,
               links=NA,
               names.links=NA,
               images=NA,
               names.images=NA,
               band.info = NA,
               returnVl = TRUE,
               saveFile = FALSE,
               saveName="MapObj.RData")
```

Arguments

file	path name of file
obj	object of the class marrayInfo, marrayRaw or marrayNorm
adf	annotated data frame object
df	data frame object
spot.ID	column in file or object that represents spot.IDs i.e. clone, BAC
chrom	column in file or object that represents the chromosome location
chrom.levels	vector indicating how the chrom column is represented. i.e chr1, chrom1, 1, etc.
base.chrm	vector of characters to be replaced in chromosome to achieve numeric vector
reg.exp	logical indication if base.chrm entry is exact match or should be used in perl regular expression mapping
loc	column in file or object representing genomic location. See details for more information
loc.start	column in file or object representing starting genomic location. See details for more information
loc.stop	column in file or object representing ending genomic location. See details for more information
locBy	either within or across to represent if the genomic locations are within, with respect to, chromosome or across the entire genomic.
file.sep	seperation character for file
additional	indication of additional columns in file or object that should be included in mapping object
names.additional	optional header names for additional columns to be included
links	indication of columns in file or object that represent hyperlinks, character or numeric, or a data.frame, with rows equal to the number of rows of the file, of hyperlinks that must be in the same order as they appear in the file
names.links	optional header names for links
images	indication of columns in file or object that represent images, character or numeric, or a data.frame, with rows equal to the number of rows of the file, of images that must be in the same order as they appear in the file
names.images	optional header names for images
band.info	a band.info object. see makeBandInfo
returnVl	logical indicating if object created should be returned
saveFile	logical indicating if object created should be saved to a file
saveName	if saveVl, path name of file
...	additional arguments to the read.table function for reading the file

Details

The mapping file must minimally contain columns for name, chromosome, and genomic location. The genomic location may be given by a single value assumed to be a central location and should be specified using the loc argument. This value should be the genomic location with respect to the entire genome not the location within a chromosome. The other, recommended, way of specifing the genomic location is with both a starting and stopping genomic location, and should be specified using loc.start and loc.stop arguments. These values are the location within the chromosome not over the entire genome. If loc.start and loc.stop are used, loc should be NA.

The band.info argument is an object of the class band.info. This object contains information about chromosome, arm, broad.band, and fine.band regions. See band.info help files for details.

The iGGV function has the capabilities of including hyperlinks in displayed tool-tips. In order for the links to display properly they need to be specified in the mapping object. If there are links in the file, the column number or head name may be specified as links. The user may add their own hyperlinks by assigning the links argument to a table of the dimension n by m where n is equivalent to the number of items in the original file. The links must be in proper format of a valid web address, i.e http://completewebpathhere.com . Similarly, images may be displayed using the images argument.

Value

an object of the class mapobj which contains mapping and band information

Author(s)

Lori A. Shepherd

See Also

mapobj, mapping.info, bandinfo, iGGV

Examples

```
#
# The following code will build the same object as data(mapping.info)
#

library("iGenomicViewer")


# writes out text files to use
writeExFiles()

# loads a band.info object
data(Band.Info)

# make mapping information object
mapping.info = mappingObj(file="HB19Kv2.HG18.txt", spot.ID="Clone", chrom="Chromosome",  chrom.levels=c("chr1","
```

TIplot-class *TIplot 'tiled image' class*

Description

The tiled image object contains all necessary information to plot a tiled image heatmap. This includes but is not limited to matrix of values, axis ranges, axis labels, size, and mapping information.

TIplot structure

tractBound The tractBound object contains all information for breaking up the yaxis spot list into different tracks. For detailed information please see vignette

vls The vls list contains information on values for plotting. There is a matrix or start and stop genomic locations for the y axix, Y. There is also the matrix of values, Z. The object H is a numeric indicating the number of tracks to break each sample into. Xcoords is a numeric list indicating each tracks x value. Finally nsmp and nBAC indicate the number of samples and the number of spots used. For more detail see vignette.

lims The lims list contains vectors of the range for the xaxis, yaxis and Z matrix [xlim, ylim, zlim]

labels The labels list contains values for individual x labels and ylabels, as well as labels for title, xaxis and yaxis. [ttl, xlab, ylab]

Zcol Zcol is a vector of colors for the Z matrix scale

cex The cex, or size, list consists of the x axis and y axis values for point size, given as xcex and ycex respectively

map The map list consists of the spot ID index [bacDX] and the associated mapping object [mapObj]. Please see mapobj for details on this structure

Note

Users are encouraged to view the vignette for more detail

Author(s)

Lori A. Shepherd, Daniel P. Gaile

See Also

initTiled

updateGGV *Updates a 'GGV' object with new known track regions*

Description

This fucntion will update a GGV object with new known regions of interest.

Usage

```
updateGGV(GGV,
          trackRegions,
          appendTo=TRUE,
          returnVl=TRUE,
          saveFlag=FALSE,
          saveName="GGVobj.RData")
```

Arguments

GGV	A GGV object. See initGGV
trackRegions	A object made by or with the same structure as the makeTrack function. See makeTrack
appendTo	Logical indicating if the regions should be added to the existing trackRegion object or if the new regions should totally replace the GGV object trackRegions
returnVl	Should GGV object be returned
saveFlag	Should GGV object be saved
saveName	If saveFlag, path file name to save object

Details

This function acts on an exisiting GGV object. The user can either replace the known regions in the GGV with new regions, or append new regions to the existing GGV trackRegion object.

For more information on trackRegion see makeTrack help file or package vignette.

Value

If returnVl, a new updated GGV object is returned.

Note

see vignette for more details

Author(s)

Lori A. Shepherd

See Also

initGGV, makeTrack, GGVobj

Examples

```
# load library
library("iGenomicViewer")

# load example data
data(iGGVex)
data(mapping.info)
data(annObj)


mapObj = mapping.info
chrArm = c("8p", "18p")
y.lbls$Pdisc = round(y.lbls$Pdisc,3)
pvls = rep(rep(rep(1:4, each=5), 960))[1:length(mapObj$mapping.info$g.loc.center)]
plot.vec = pvls[1:length(mapObj$mapping.info$g.loc.center)]
side.plot.extras="title(main='pvals'"

# make track Region
trackRegion = makeTrack(Fine.Band = c("8p11.22","18p11.21"),genomicLoc = NA, geneName = "FANCE")


# make GGV object
GGV = initGGV(vls = round(mat, 3),
              mapObj = mapObj,
              annObj = annObj,
              x.labels=x.lbls,
              y.labels=y.lbls,
              xy.labels=NA,
              chrArms=chrArm,
              trackRegions=trackRegion,
              side.plot.extras=side.plot.extras,
              plot.vec=plot.vec,
              plot.dx=which(mapObj$mapping.info$Chrom=="chr8"),
              mapObj.columns =c(2,8,10),
              smpLines=TRUE,
              divCol="lightgrey")


# make new track region
trackRegionUpdate = makeTrack(geneName="MYC")


# this will append new regions to old object regions
GGVnew = updateGGV(GGV=GGV,
                   trackRegions=trackRegionUpdate,
                   appendTo=TRUE)
```

```
# this will replace GGV object's trackregions with
# new object
GGVnew = updateGGV(GGV=GGV,
                   trackRegions=trackRegionUpdate,
                   appendTo=FALSE)
```

---

writeExFiles *Writes Sample Files*

Description

This function writes example annotation, mapping, and band information files.

Usage

```
writeExFiles(direct="./")
```

Arguments

direct	directory path where files will be written

Details

This function writes out sample annotation, mapping, and band information files. see individual help files for more details CancerGenes cytoBand DiseaseGenes DNArepairgenes HB19Kv2.HG18

Value

Five files are written to the given directory

Author(s)

Lori A. Shepherd

See Also

annObj,Band.Info, mapping.info

Examples

```
library("iGenomicViewer")

# writes to current working directory
writeExFiles()
```

Index

∗Topic **classes**
anninfo-class, 3
bandinfo-class, 8
GGVobj-class, 13
mapobj-class, 40
TIplot-class, 45
∗Topic **datasets**
annObj, 4
Band.Info, 7
CancerGenes, 9
cytoBand, 11
DiseaseGenes, 12
DNArepairgenes, 13
HB19Kv2.HG18, 15
iGGVex, 21
mapping.info, 40
∗Topic **iplot**
iGGV, 16
iGGVtiled, 21
∗Topic **methods**
annotation, 4
convertLoc, 9
iGGV, 16
iGGVtiled, 21
initGGV, 26
initTile, 30
makeBandInfo, 32
makeGGV, 34
makeTiled, 37
makeTrack, 38
mappingObj, 41
updateGGV, 46
writeExFiles, 48
∗Topic **package**
iGenomicViewer-package, 2

anninfo, *6*
anninfo *(*anninfo-class*)*, 3
anninfo-class, 3
annObj, 4, *6*, *48*
annobj *(*annObj*)*, 4
annotation, *3*, 4, *20*
annotationObj *(*annotation*)*, 4

Band.Info, 7, *8*, *10*, *34*, *48*
Band.info *(*Band.Info*)*, 7
band.info *(*Band.Info*)*, 7
BandInfo *(*bandinfo-class*)*, 8
bandinfo, *6*, *34*, *44*
bandinfo *(*bandinfo-class*)*, 8
bandinfo-class, 8

CancerGenes, 9
convertCloc *(*convertLoc*)*, 9
convertGloc *(*convertLoc*)*, 9
convertLoc, 9, *32*
cytoBand, *8*, 11
cytoband *(*cytoBand*)*, 11

DiseaseGenes, 12
DNArepairGenes *(*DNArepairgenes*)*, 13
DNArepairgenes, 13

GGVobj, *29*, *36*, *47*
GGVobj *(*GGVobj-class*)*, 13
GGVobj-class, 13

HB19Kv2.HG18, 15
HB19Kv2HG18 *(*HB19Kv2.HG18*)*, 15

iGenomicViewer
*(*iGenomicViewer-package*)*, 2
iGenomicViewer-package, 2
iGGV, *3*, *6*, 16, *25*, *29*, *34*, *44*
iGGVex, *20*, 21
iGGVtiled, *3*, 21, *32*, *37*
initGGV, *15*, 26, *36*, *47*
initTile, *25*, 30, *37*
initTiled, *45*
initTiled *(*initTile*)*, 30

makeAnnotation *(annotation)*, 4
makeBandInfo, *8*, 32
makeGGV, *15*, 34, *39*
makeTile, *32*
makeTile *(makeTiled)*, 37
makeTiled, *25*, 37
makeTrack, *29*, 38, *47*
mapObj *(mapobj-class)*, 40
mapobj, *44*
mapobj *(mapobj-class)*, 40
mapobj-class, 40
mapping.info, 40, *44*, *48*
mappingObj, *20*, *40*, 41
mappingObjADF *(mappingObj)*, 41
mappingObjDF *(mappingObj)*, 41
mappingObjMarray *(mappingObj)*, 41
mat *(iGGVex)*, 21

TIplot, *32*
TIplot *(TIplot-class)*, 45
TIplot-class, 45
trackRegion *(makeTrack)*, 38
trackRegions *(makeTrack)*, 38

updateGGV, *15*, *39*, 46

writeExFiles, 48

x.lbls *(iGGVex)*, 21

y.lbls *(iGGVex)*, 21