Package ‘sendplot’

February 20, 2015

Version 4.0.0

Date March 01, 2013

Title Tool for sending interactive plots 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), rtiff

Suggests stats

SystemRequirements libtiff

Description A tool for visualizing data

LazyData no

License GPL (>= 2)

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

NeedsCompilation no

Repository CRAN

Date/Publication 2013-04-25 07:35:55

R topics documented:

`sendplot-package` *sendplot package*

Description

The sendplot package creates an interactive layout of plots that may be viewed in a web browser.

Details

Users are encouraged to view vignette for more details and several useful examples.

The user must initialize a Splot object (`initSplot`) Then the user may optionally add interactive regions to any of the figures in the layout. (`makeImap`, `addDefault`) A static image (postscript, png or jpeg), and/or an interactive HTML file is created (`makeSplot`)

Note

The interactive html plot currently only works in web browsers that implement java script.

The code used to create the javascript embedded in html file is a modified version of the javascript code or from the open source tooltip library. see reference links

Author(s)

Lori A. Shepherd, Daniel P. Gaile, Lara Sucheston, Andrew Bruno, Kenneth F. Manly

References

http://www.R-project.org

http://www.onlamp.com/pub/a/onlamp/2007/07/05/writing-advanced-javascript.html

http://www.walterzorn.com/tooltip/tooltip_e.htm

See Also

`initSplot`, `makeImap`, `makeSplot`, `sendplot`, `layout`

`aCGHex` *DATA FOR VIGNETTE EXAMPLE*

Description

This data object contains information needed for running vignette example.

Format

An aCGHplus object

Details

This file contains a data object from the R package aCGHplus. aCGHplus is a package designed for array comparative genomic hybridization experiments. For information on this package and objects that can be created with this package please go to the website: http://sphhp.buffalo.edu/biostat/research/software/acghplus/ind This is a data set of 10 samples. The mapping.info object and log2 matricies have been altered to only store information for the region 4q13; aCGH objects tend to be large, we opted to only include data needed for running the vignette example..

Note

This is included for running vignette example

Source

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

References

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

addBounding *Adds Markers For Auto-detection of Bounding Points*

Description

NOT CALLED BY USER. The addBounding function is utilized by the makeImap function to create a file with bounding locations for a desired interactive plot.

Usage

```
addBounding(Splot,
            figure,
            bb.clr = "blue",
            bb.cex = 2,
            boundFileName = "SplotDot",
            dir="./")
```

Arguments

`Splot`	An Object of the class Splot
`figure`	The numeric indication for which figure's bounding box points should be displayed
`bb.clr`	color of bounding points
`bb.cex`	size of bounding points
`boundFileName`	name to use for .png file name
`dir`	directory path to where files should be created

Details

This function adds phantom points to a figure to determine a plots bounding box limits.

Value

A .png file is created with points at a figures plotting box bounds. This file is used for automatic mapping of points

Note

NOT CALLED BY USER. INTERNAL HELPER FUNCTION

Author(s)

Lori A.Shepherd, Daniel P. Gaile

See Also

makeImap

Examples

```
# not called by user
```

addDefault ADDS DEFAULT TOOL-TIP REGION

Description

The default region in an html image map is any part of the figure not already specified with a different region (i.e. rect, circle, poly). This function adds tool-tip information to the default region.

Usage

```
addDefault(Splot,
          data=NA,
          data.labels=NA,
          links=NA,
          links.labels=NA,
          asLink=NA,
          font.type="Helvetica",
          font.color="black",
          font.size="12",
          bg.color='#D6E3F6',
          returnVl=TRUE,
          saveFlag=FALSE,
          saveName="Splot.RData")
```

Arguments

Splot	An Object of the class Splot
data	character vector containing data to be displayed in tool-tip
data.labels	names descripting data vector
links	character vector containing complete web address for hyperlinks within tool-tip
links.labels	names describing links vector
asLink	complete web address for area to be treated as hyperlink
font.type	font type for tool-tip. Currently support fonts are Arial, Helvetica,and sans-serif
font.color	font color for tool-tip
font.size	font size in tool-tip

`bg.color`	background color of tool-tip
`returnVl`	logical indicating if Splot object should be returned
`saveFlag`	logical indicating if Splot object should be saved to a file
`saveName`	if saveFlag, path and file name to save object

Details

This function takes in data and links vectors, and converts into proper syntax for tool-tip. It also sets up default tool-tip display regarding font color, size, and type, as well as background color.

Value

The Splot object is updated to include a Default.Obj containing vectors in proper syntax and tool-tip display information.

Note

There can only be one default for a given layout of figures. If this is run more than once it will override previous calls.

Author(s)

Lori A.Shepherd, Daniel P. Gaile

References

http://www.onlamp.com/pub/a/onlamp/2007/07/05/writing-advanced-javascript.html

http://www.walterzorn.com/tooltip/tooltip_e.htm

See Also

`initSplot`

Examples

```
library(sendplot)
library(rtiff)

# sets up matrix for layout
mat = matrix(1,nrow=12, ncol=13)
mat[9:12,] = 2
mat[,9:13] = 3
mat[1:2,] = 4

# sets up figure margins
mai.mat = matrix(.5,ncol=4,nrow=4)

# vector of image values
myX = c(-1,-10,1,10,-5,0)
```

```
# plot calls
plot.calls = c("boxplot(count ~ spray, data = InsectSprays, col = 'lightgray')", "plot(1:3,1:3, col='blue', xlab=
plt.extras=list(figure1= "rect(xleft=c(3,1), ytop=c(25,5),xright=c(4,2),  ybottom=c(20,0));title(main='A', cex=


# initialize Splot object
Splot = initSplot(mat, plot.calls, mai.mat = mai.mat,plot.extras =plt.extras)

# add default
Splot = addDefault(Splot, data=c("This is default", "data2"), data.labels=c("label", "d2"), links=c("http://www.



#set up temporary directory
direct = paste(tempdir(),"/",sep="")
direct


# make plot that has interactive default
Splot = makeSplot(Splot, fname.root="exToy",dir=direct, returnObj=TRUE)
```

automapPts *Compares Images To Find Bounding Coordinates In Pixils*

Description

NOT CALLED BY USER. The automapPts function is utilized by makeImap to retrieve the up.left and lower.right pixil coordinates of a desired interactive figure

Usage

```
automapPts(Splot,
           fname.root="Splot",
           boundFileName="SplotDot",
           dir="./",
           automap.method="mode")
```

Arguments

Splot	An object of the class Splot
fname.root	Base name of the static version of plots
boundFileName	Base name of the version of plots with the given figure's additional bounding points displayed
dir	directory path to where files were created
automap.method	Method to detect upper and lower bounds. Current options are "mode" or "median"

Details

This function retrieves the up.left and lower.right pixil coordinates of a desired interactive figure. The function assumes that a 'clean' image without additional bounding points and a 'bounding' image that has additional bounding points (see `addBounding`) are created. It converts these images to .tif files and uses the rtiff package readTiff to compare the images for differences, thus picking up the additional points.

Value

A list with up.left and low.right bounding coordinates in pixils or NA if could not map correctly

Note

NOT CALLED BY USER. INTERNAL HELPER FUNCTION

utilizes readTiff from rtiff library

Author(s)

Lori A.Shepherd, Daniel P.Gaile

References

Eric Kort (2006). rtiff: A tiff reader for R.. R package version 1.1.

See Also

`makeImap`, `rtiff`, `getBounds`, `mapMethod`

Examples

```
# not called by user
```

`eval.js` *A JAVASCRIPT-LIKE EVAL FUNCTION*

Description

This function evaluates expressions.

Usage

```
eval.js(expr,
        envir=parent.frame(),
        enclos=if(is.list(envir)||is.pairlist(envir)) parent.frame())
```

Arguments

`expr`	character string of an expression to evaluate
`envir`	passed to eval function. see eval
`enclos`	passed to eval function. see eval

Details

Wrapper to the eval functions from the R base package. Evaluates character expression.

Value

will return the evaluated expression

Note

uses function eval from base package

Author(s)

James Java

References

R base package function eval

See Also

`eval`

Examples

```
df = list()
df$a = rep(1,5)
df$b = rep(”one”,5)
df = as.data.frame(df)

#for comparison view
df

eval.js(”df$new = NA”)
df
```

getBounds *Compares Two Tif Images To Find Where They Differ*

Description

NOT CALLED BY USER. The getBounds function is utilized by the automapPts function to compare two tif images for differences

Usage

```
getBounds(channelClr,
          tif.fin,
          tif.dot,
          automap.method="mode")
```

Arguments

`channelClr`	Which color channel to compare: "red", "blue", "green"
`tif.fin`	Path file name to tif image of plots without additional bounding point
`tif.dot`	Path file name to tif image with the given figure's additional bounding points displayed
`automap.method`	Method to detect upper and lower bounds. Current options are "mode" or "median"

Details

This function is used by the automapPts function to retrieve up.left and low.right pixil coordinates of a figure. getBounds will check a certain channel (red, green, or blue) of two rtiff/pixmap pixmapRGB objects. It creates a logical matrix 0 if equal and 1 if different. This matrix gets passed into the mapMethod function to determine location.

Value

A list with up.left and low.right bounding coordinates in pixils or NA if could not map correctly

Note

NOT CALLED BY USER. INTERNAL HELPER FUNCTION

Author(s)

Lori A.Shepherd, Daniel P. Gaile

See Also

`makeImap`, `rtiff`, `pixmap`, `automapPts`, `mapMethod`

Examples

```
# not called by user
```

---

heatmap.send *INTERACTIVE HEATMAP*

Description

This function is a wrapper for the R stats package heatmap. This will create an interactive heatmap image. NOTE: The majority of the code for this function is verbatim from the R package stats heatmap function. This function was designed to work as a wrapper to untilize the same functionality and plotting as the heatmap function with sendplot's interactive functionality.

Usage

```
heatmap.send(x,Rowv = NULL,
             Colv = if (symm) "Rowv" else NULL,
             distfun = dist,hclustfun = hclust,
             reorderfun = function(d,w) reorder(d, w),
             add.expr,symm = FALSE,
             revC = identical(Colv,"Rowv"),
             scale = c("row", "column", "none"),
             na.rm = TRUE, margins = c(5, 5),
             ColSideColors,RowSideColors,
             MainColor = heat.colors(12),
             cexRow = 0.2 +  1/log10(nr),
             cexCol = 0.2 + 1/log10(nc),
             labRow = NULL,labCol = NULL,
             main = NULL,xlab = NULL,ylab = NULL,
             keep.dendro = FALSE,
             verbose = getOption("verbose"),

             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,
             spot.radius=5,source.plot=NA,
             image.size="800x1100",
             fname.root="test",dir="./", header="v3",
             window.size = "800x1100",
             ...)
```

Arguments

`x` numeric matrix of the values to be plotted

`Rowv` determines if and how the row dendrogram should be computed and reordered. Either a ’dendrogram’ or a vector of values used to reorder the row dendrogram or ’NA’ to suppress any row dendrogram (and reordering) or by default, ’NULL’, see heatmap argument

`Colv` determines if and how the column dendrogram should be reordered. Has the same options as the ’Rowv’ argument above and additionally when ’x’ is a square matrix, ’Colv = "Rowv"’ means that columns should be treated identically to the rows

`distfun` function used to compute the distance (dissimilarity) between both rows and columns. Defaults to ’dist’

`hclustfun` function used to compute the hierarchical clustering when ’Rowv’ or ’Colv’ are not dendrograms. Defaults to ’hclust’

`reorderfun` function(d,w) of dendrogram and weights for reordering the row and column dendrograms. The default uses ’reorder.dendrogram’

`add.expr` expression that will be evaluated after the call to ’image’. Can be used to add components to the plot

`symm` logical indicating if ’x’ should be treated *symm*etrically; can only be true when ’x’ is a square matrix.

`revC` logical indicating if the column order should be ’rev’ersed for plotting, such that e.g., for the symmetric case, the symmetry axis is as usual

`scale` character indicating if the values should be centered and scaled in either the row direction or the column direction, or none. The default is ’"row"’ if ’symm’ false, and ’"none"’ otherwise

`na.rm` logical indicating whether ’NA”s should be removed

`margins` numeric vector of length 2 containing the margins (see ’par(mar= *)’) for column and row names, respectively

`ColSideColors` (optional) character vector of length ’ncol(x)’ containing the color names for a horizontal side bar that may be used to annotate the columns of ’x’

`RowSideColors` (optional) character vector of length ’nrow(x)’ containing the color names for a vertical side bar that may be used to annotate the rows of ’x’

`MainColor` color scale for values. Passed into ’image’ function as col argument

`cexRow` positive number, used as ’cex.axis’ in for the row axis labeling. The defaults currently only use number of rows

`cexCol` positive number, used as ’cex.axis’ in for the column axis labeling. The defaults currently only use number of columns

`labRow` character vectors with row labels to use; these default to ’rownames(x)’

`labCol` character vectors with column labels to use; these default to ’colnames(x)’

`main` main title; defaults to none

`xlab` x axis title; defaults to none

`ylab` y axis title; defautls to none

`keep.dendro` logical indicating if the dendrogram(s) should be kept as part of the result (when 'Rowv' and/or 'Colv' are not NA)

`verbose` logical indicating if information should be printed

`x.labels` data frame of n x m which contains values relating to the x axis of the heatmap plot. n should be equal to the second dimension of the x argument.This information is displayed in the interactive plot window. This may be left as NA.

`y.labels` data frame of n x m which contains values relating to the y axis of the heatmap plot. n should be equal to the first dimension of the x argument.This information is displayed in the interactive plot window. This may be left as NA

`xy.labels` list of matricies. All matricies should be of n x m where n is equal to the first dimension of the x argument and m is equal to the second dimension of the x argument. This information is displayed in the interactive plot window. This may be left NA

`x.links` data frame of n x m which contains web addresses for links relating to the x axis of the heatmap plot. n should be equal to the second dimension of the x argument. m columns contains information regarding sample. This information is displayed as hyperlinks in the interactive plot window. This may be left NA

`y.links` data frame of n x m which contains web addresses for links relating to the y axis of the heatmap plot. n should be equal to the first dimension of the x argument. This information is displayed as hyperlinks in the interactive plot window. This may be left as NA

`xy.links` list of matricies. All matricies should be of n x m where n is equal to the first dimension of the x argument and m is equal to the second dimension of the x argument. 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. May be a data.frame or matrix of n x m where n is equal to the first dimension of the x argument and m is equal to the second dimension of the x argument, a vector of length equal to the first dimension of the x argument that will be repeated, a vector of length equal to the second dimension of the x argument that will be repeated,a non NA value of length 1 that will be repeated for all points, or a vector of length dim(x)[1]*dim(x)[2]

`x.images` data frame of n x m which contains paths for images relating to the x axis of the heatmap plot. n should be equal to the second dimension of the x argument. m columns contains information regarding sample. This information is displayed as images in the interactive plot window. This may be left NA

`y.images` data frame of n x m which contains paths for images relating to the y axis of the heatmap plot. n should be equal to the first dimension of the x argument. This information is displayed as images in the interactive plot window. This may be left as NA

`xy.images` list of matricies. All matricies should be of n x m where n is equal to the first dimension of the x argument and m is equal to the second dimension of the x argument. This information is displayed in the interactive plot window as images. The values in these matricies should be complete path of images

`spot.radius` radius of circle in pixels indicating area that will be interactive around the center of graphed points `source.plot` Indicates whether application should make a postscript file and then convert to png file, or if the png file should be made directly. This value is either ps, png, or NA. If NA the operating system is checked and the appropraite file format is output. Unix has a convert function that can convert a ps file to png file; we by default use this setup because we feel the postscript file maintains better quality. So on unix/linux systems if source.plot is NA, source.plot will be set to ps. Windows does not have this option, for this reason source.plot will be set to png if left NA

`image.size` character indicating size of device.

`fname.root` Base name to use for all files created.

`dir` directory path to where files should be created. Default creates files in working directory

`header` May either be v1,v2, or v3. This determines which tooltip header will be in the html file. Each version has different features or works well with different web browsers. see sp.header for details.

`window.size` size of the html window. Only effective when header=v3

`...` additional arguments to the makeImap function

Details

The majority of the code for this function is verbatim from the R package stats heatmap function. This function was designed to work as a wrapper to untilize the same functionality and plotting as the heatmap function with sendplot's interactive functionality. See `heatmap` for more details on arguments and details concerning the creatation of plots.

See `sendplot` for more information regarding the creation of the interactive output with tool-tip content.

Users are encouraged to read the package vignette which includes a detailed discussion of all function arguments as well as several useful examples.

Value

creates the static and interactive versions of heatmap

Note

The majority of the code for this function is verbatim from the R package stats heatmap function. This function was designed to work as a wrapper to untilize the same functionality and plotting as the heatmap function with sendplot's interactive functionality.

The interactive html plot currently only works in web browsers that implement java script.

The code used to create the javascript embedded in html file is a modified version of the javascript code or from the open source tooltip library. see reference links

Author(s)

Lori A. Shepherd and Daniel P. Gaile;

Authors of heatmap code used in our code: Andy Liaw, original; R. Gentleman, M. Maechler, W. Huber,revisions

References

http://www.R-project.org

http://www.onlamp.com/pub/a/onlamp/2007/07/05/writing-advanced-javascript.html

http://www.walterzorn.com/tooltip/tooltip_e.htm

See Also

`initSplot`,`makeImap`,`makeSplot`,`imagesend`,`heatmap.send.legacy`, `sendplot`, `heatmap`

Examples

```
library(sendplot)
library(rtiff)

require(graphics)


x  = as.matrix(mtcars)
rc = rainbow(nrow(x), start=0, end=.3)
cc = rainbow(ncol(x), start=0, end=.3)

xy.labels=list(value=x)


x.labels=data.frame(label=colnames(x),
  description=c("Miles/(US) gallon","Number of cylinders",
    "Displacement (cu.in.)",
    "Gross horsepower",
    "Rear axle ratio",
    "Weight (lb/1000)",
    "1/4 mile time",
    "V/S",
    "Transmission (0 = automatic, 1 = manual)",
    "Number of forward gears",
    "Number of carburetors")
  )


#set up temporary directory
direct = paste(tempdir(),"/",sep="")
direct


heatmap.send(x,scale="column", xy.labels = xy.labels,
                 x.labels=x.labels,
                 RowSideColors = rc, ColSideColors = cc, margin=c(5,10),
                 xlab = "specification variables", ylab= "Car Models",
                 main = "mtcars data",
                 fname.root="exHeat",dir=direct,
                 font.size=18,image.size="600x900")
```

HeatmapWrapper *INTERACTIVE HEATMAP - DEPRECATED*

Description

This function is a wrapper for the R stats package heatmap. This will create an interactive heatmap image. NOTE: The majority of the code for this function is verbatim from the R package stats heatmap function. This function was designed to work as a wrapper to untilize the same functionality and plotting as the heatmap function with sendplot's interactive functionality.

Usage

```
heatmap.send.legacy(x,Rowv = NULL,
             Colv = if (symm) "Rowv" else NULL,
             distfun = dist,hclustfun = hclust,
             reorderfun = function(d,w) reorder(d, w),
             add.expr,symm = FALSE,
             revC = identical(Colv,"Rowv"),
             scale = c("row", "column", "none"),
             na.rm = TRUE, margins = c(5, 5),
             ColSideColors,RowSideColors,
             cexRow = 0.2 +  1/log10(nr),
             cexCol = 0.2 + 1/log10(nc),
             labRow = NULL,labCol = NULL,
             main = NULL,xlab = NULL,ylab = NULL,
             keep.dendro = FALSE,
             verbose = getOption("verbose"),
             mai.mat=NA, mai.prc=FALSE,
             z.value="value",
             x.lbls=NA,y.lbls=NA,xy.lbls=NA,
             x.links=NA, y.links=NA,
             xy.links=NA,asLinks=NA,
             bound.pt = FALSE, source.plot=NA,
             resize="800x1100",
             ps.paper="letter",ps.width=8,ps.height=11,
             fname.root="test",dir="./", header="v2",
             paint=FALSE, img.prog = NA,
             up.left=c(288,203),low.right=c(620,940),
             spot.radius=5, automap=FALSE, automap.method="mode")
```

Arguments

`x` numeric matrix of the values to be plotted

`Rowv` determines if and how the row dendrogram should be computed and reordered. Either a ’dendrogram’ or a vector of values used to reorder the row dendrogram or ’NA’ to suppress any row dendrogram (and reordering) or by default, ’NULL’, see heatmap argument

`Colv` determines if and how the column dendrogram should be reordered. Has the same options as the ’Rowv’ argument above and additionally when ’x’ is a square matrix, ’Colv = "Rowv"’ means that columns should be treated identically to the rows

`distfun` function used to compute the distance (dissimilarity) between both rows and columns. Defaults to ’dist’

`hclustfun` function used to compute the hierarchical clustering when ’Rowv’ or ’Colv’ are not dendrograms. Defaults to ’hclust’

`reorderfun` function(d,w) of dendrogram and weights for reordering the row and column dendrograms. The default uses ’reorder.dendrogram’

`add.expr` expression that will be evaluated after the call to ’image’. Can be used to add components to the plot

`symm` logical indicating if ’x’ should be treated *symm*etrically; can only be true when ’x’ is a square matrix.

`revC` logical indicating if the column order should be ’rev’ersed for plotting, such that e.g., for the symmetric case, the symmetry axis is as usual

`scale` character indicating if the values should be centered and scaled in either the row direction or the column direction, or none. The default is ’"row"’ if ’symm’ false, and ’"none"’ otherwise

`na.rm` logical indicating whether ’NA’’s should be removed

`margins` numeric vector of length 2 containing the margins (see ’par(mar= *)’) for column and row names, respectively

`ColSideColors` (optional) character vector of length ’ncol(x)’ containing the color names for a horizontal side bar that may be used to annotate the columns of ’x’

`RowSideColors` (optional) character vector of length ’nrow(x)’ containing the color names for a vertical side bar that may be used to annotate the rows of ’x’

`cexRow` positive number, used as ’cex.axis’ in for the row axis labeling. The defaults currently only use number of rows

`cexCol` positive number, used as ’cex.axis’ in for the column axis labeling. The defaults currently only use number of columns

`labRow` character vectors with row labels to use; these default to ’rownames(x)’

`labCol` character vectors with column labels to use; these default to ’colnames(x)’

`main` main title; defaults to none

`xlab` x axis title; defaults to none

`ylab` y axis title; defautls to none

`keep.dendro` logical indicating if the dendrogram(s) should be kept as part of the result (when 'Rowv' and/or 'Colv' are not NA)

`verbose` logical indicating if information should be printed

`mai.mat` n x 4 matrix of values to be passed in for each plots par mai. n is equal to the length of plot.calls. If NA, uses default margins. Utitilzing this wrapper n will either be 3,4, or 5.

`mai.prc` logical indicating if mai mat values are percentages or hard coded values. If mai.proc is T, indicates percentage.

`z.value` character vector indicating the label for what the z argument holds.

`x.lbls` data frame of n x m which contains values relating to the x axis of the heatmap plot. n should be equal to the second dimension of the x argument.This information is displayed in the interactive plot window. This may be left as NA.

`y.lbls` data frame of n x m which contains values relating to the y axis of the heatmap plot. n should be equal to the first dimension of the x argument.This information is displayed in the interactive plot window. This may be left as NA

`xy.lbls` list of matricies. All matricies should be of n x m where n is equal to the first dimension of the x argument and m is equal to the second dimension of the x argument. This information is displayed in the interactive plot window. This may be left NA

`x.links` data frame of n x m which contains web addresses for links relating to the x axis of the heatmap plot. n should be equal to the second dimension of the x argument. m columns contains information regarding sample. This information is displayed as hyperlinks in the interactive plot window. This may be left NA

`y.links` data frame of n x m which contains web addresses for links relating to the y axis of the heatmap plot. n should be equal to the first dimension of the x argument. This information is displayed as hyperlinks in the interactive plot window. This may be left as NA

`xy.links` list of matricies. All matricies should be of n x m where n is equal to the first dimension of the x argument and m is equal to the second dimension of the x argument. 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. May be a data.frame or matrix of n x m where n is equal to the first dimension of the x argument and m is equal to the second dimension of the x argument, a vector of length equal to the first dimension of the x argument that will be repeated, a vector of length equal to the second dimension of the x argument that will be repeated,a non NA value of length 1 that will be repeated for all points, or a vector of length dim(x)[1]*dim(x)[2]

`bound.pt` logical indicating if blue points should be plotted to aid in finding the upper left and lower right coordinates of a hte heatmap. If bound.pt is FALSE, indicates that up.left and low.right arguments are correct and will make the html file

`source.plot` Indicates whether application should make a postscript file and then convert to png file, or if the png file should be made directly. This value is either ps, png, or NA. If NA the operating system is checked and the appropraite file format is output. Unix has a convert function that can convert a ps file to png file;

we by default use this setup because we feel the postscript file maintains better quality. So on unix/linux systems if source.plot is NA, source.plot will be set to ps. Windows does not have this option, for this reason source.plot will be set to png if left NA

`resize` character indicating resize value. The postscript version will be resized to this value when converted to .png.

`ps.paper` postscript paper argument

`ps.width` poscript width argument

`ps.height` postscript height argument

`fname.root` Base name to use for posctscript, .png, and html file names.

`dir` directory path to where files should be created. Default creates files in working directory

`header` May either be v1 or v2. This determines which tooltip header will be in the html file. Each version has different features or works well with different web browsers. see sp.header for details.

`paint` logical indicating if application should automatically open .png file for the user to view .png file and/or to retrieve needed bounding values of the first plot call. see details

`img.prog` If paint is TRUE, the command line call that will open a program to view .png file to retrieve pixil locations of interactive plot bounds. If this is left NA, the operating system is checked and a default program is used. For unix the default application is kolourpaint and for windows it is microsoft paint (mspaint)

`up.left` The x and y value in pixels of the upper left hand corner of the first plot call. see details

`low.right` The x and y value in pixels of the lower right hand corner of the first plot call. see details

`spot.radius` radius of circle in pixels indicating area that will be interactive around the center of graphed points

`automap` automatic detection of up.left and low.right bound points. Fully functional on linux/unix machines only.

`automap.method` Method to detect upper and lower bounds. Current options are mode or median

Details

The majority of the code for this function is verbatim from the R package stats heatmap function. This function was designed to work as a wrapper to untilize the same functionality and plotting as the heatmap function with sendplot's interactive functionality. See `heatmap` for more details on arguments and details concerning the creatation of plots.

See `sendplot` for more information regarding the creation of the interactive output with tool-tip content.

Users are encouraged to read the package vignette which includes a detailed discussion of all function arguments as well as several useful examples.

Value

Creates a static .ps and .png file, and an interactive html file

Note

The majority of the code for this function is verbatim from the R package stats heatmap function. This function was designed to work as a wrapper to untilize the same functionality and plotting as the heatmap function with sendplot's interactive functionality.

The x and y mappings to the interactive plot are created using the x and y vectors passed in as an argument to sendplot. Note: this could be handy if for example the user plotted more points to the first plot using the plt.extras argument. If the user wanted all points interactive, the x and y values of the sendplot argument would be a combination of all plotted points.

The interactive html plot currently only works in web browsers that implement java script.

The code used to create the javascript embedded in html file is a modified version of the javascript code or from the open source tooltip library. see reference links

This function is deprecated. Please see heatmap.send for updated version.

Author(s)

Lori A. Shepherd and Daniel P. Gaile;

Authors of heatmap code used in our code: Andy Liaw, original; R. Gentleman, M. Maechler, W. Huber,revisions

References

http://www.R-project.org

http://www.onlamp.com/pub/a/onlamp/2007/07/05/writing-advanced-javascript.html

http://www.walterzorn.com/tooltip/tooltip_e.htm

See Also

heatmap.send, sendplot,sendxy, sendimage, heatmap

Examples

```
rm(list=ls())
library(sendplot)

# mock matrix of values
temp = matrix(rnorm(15), nrow=5, ncol=3)

# color bars for samples
rcol = c("red", "blue", "yellow", "purple", "blue")
ccol = c("black", "green", "black")

# matrix of values to display in interactive window
temp.inv = list()
temp.inv$sample.ID = c("smp1", "smp2", "smp3")
temp.inv$count = 1:3
temp.inv$bb = 3:1
temp.inv = as.data.frame(temp.inv)

# matrix of values to display in interactive window
loc.inv = list()
loc.inv$spot = c("sp1", "sp2", "sp3","sp4","sp5")
loc.inv$vv = 1:5
loc.inv$bv = 5:1
loc.inv = as.data.frame(loc.inv)


#set up temporary directory
direct = paste(tempdir(),"/",sep="")
direct

heatmap.send.legacy(temp, RowSideColors=rcol, ColSideColors=ccol,
x.lbls=temp.inv, y.lbls=loc.inv, bound.pt=FALSE, paint=FALSE,
spot.radius=20, dir=direct)


# or display heatmap without color bands
heatmap.send.legacy(temp, x.lbls=temp.inv, y.lbls=loc.inv, bound.pt=FALSE,
paint=FALSE, spot.radius=20,dir=direct)


# or without cluster
heatmap.send.legacy(temp, Rowv=NA, Colv=NA, x.lbls=temp.inv, y.lbls=loc.inv, bound.pt=FALSE,
paint=FALSE, spot.radius=20,dir=direct)
```

imagesend *INTERACTIVE IMAGE*

Description

This function is a wrapper to sendplot that will create a single interactive image

Usage

```
imagesend(plot.call,
          x.pos,
          y.pos,
          xy.type,
          plot.extras = NA,
          mai.mat=NA, mai.prc=FALSE,
          xy.labels=NA,
          image.size="800x1100",
          spot.radius = 5,
          fname.root="Splot",
          dir="./",
          window.size = "800x1100",
          ...)
```

Arguments

`plot.call` character vector containing single plot call

`x.pos` vector of x locations for interactive points

`y.pos` vector of y locations for interactive points

`xy.type` Indication of how the xpos and ypos values should be treated.Current options are "image.midpoint","image.boundaries", or "image.box".See details or vignette.

`plot.extras` List of additional plotting calls that should be executed for the plot.

`mai.mat` 1 x 4 matrix of values to be passed in for each plots par mai. n is equal to the length of plot.calls. If NA, uses default margins.

`mai.prc` logical indicating if mai mat values are percentages or hard coded values. If mai.proc is T, indicates percentage.

`xy.labels` list of matricies. All matricies should be of n x m where n is the length of y and m is the length of x when xy.type is "image.midpoint". All matricies should be of n x m where n is the length of y -1 and m is the length of x - 1 when xy.type is "image.boundaries" or "image.box". This information is displayed in the interactive plot window

`image.size` character indicating size of device.

`spot.radius` radius of circle in pixels indicating area that will be interactive around the center of graphed points

`fname.root` Base name to use for all files created.

`dir` directory path to where files should be created. Default creates files in working directory

`window.size` size of the html window

`...` additional arguments to the makeImap function

Details

This function is a wrapper for the sendplot function to create a single interacive image. See `initSplot`, `makeImap`, and `makeSplot` for more information.

Users are encouraged to read the package vignette which includes a detailed discussion of all function arguments as well as several useful examples.

Value

Creates a static and interactive image

Note

The interactive html plot currently only works in web browsers that implement java script.

The code used to create the javascript embedded in html file is a modified version of the javascript code or from the open source tooltip library. see reference links

Author(s)

Lori A. Shepherd, Daniel P. Gaile

References

http://www.onlamp.com/pub/a/onlamp/2007/07/05/writing-advanced-javascript.html

http://www.walterzorn.com/tooltip/tooltip_e.htm

See Also

initSplot, makeImap, makeSplot, sendplot-package, sendimage

Examples

```
library(sendplot)
library(rtiff)
library(stats)

mai.mat = matrix(c(1,1,1,1), ncol=4)

carsX  = as.matrix(mtcars)
carsX <- sweep(carsX, 2, colMeans(carsX, na.rm = TRUE))
        sx <- apply(X=carsX, MARGIN=2, FUN="sd", na.rm = TRUE)
        carsX <- sweep(carsX, 2, sx, "/")

plot.call="image(x=1:dim(carsX)[2],y=1:dim(carsX)[1], z=t(carsX),axes =
FALSE, xlab = '', ylab = '');axis(1,1:dim(carsX)[2],
labels=colnames(carsX),las = 2, line = -0.5, tick = 0,cex.axis =.8);
axis(4,1:dim(carsX)[1], labels=rownames(carsX),las = 2, line = -0.5,
tick = 0,cex.axis =.65)"

xy.labels=list(value=round(carsX,3))

x.labels=data.frame(label=colnames(carsX),
  description=c("Miles/(US) gallon","Number of cylinders",
    "Displacement (cu.in.)",
    "Gross horsepower",
    "Rear axle ratio",
    "Weight (lb/1000)",
    "1/4 mile time",
    "V/S",
    "Transmission (0 = automatic, 1 = manual)",
    "Number of forward gears",
    "Number of carburetors")
  )
```

```
#set up temporary directory
direct = paste(tempdir(),"/",sep="")
direct

imagesend(plot.call=plot.call,
          x.pos= 1:dim(carsX)[2],
          y.pos= 1:dim(carsX)[1],
          xy.type = "image.midpoints",
          xy.labels=xy.labels,
          spot.radius = 5,
          fname.root="manImage", dir=direct,
          window.size = "800x1100",
          x.labels=x.labels, mai.mat=mai.mat)
```

initSplot *Creates A Sendplot 'Splot' Object*

Description

The initSplot function creates a sendplot 'Splot' object. A Splot object holds all necessary elements to make a static layout of images and, through other functions, all elements to make any of those images interactive with tool-tip display content

Usage

```
initSplot(mat,
          plot.calls,
          Iflag=NA,
          figTypes=NA,
          mai.mat=NA,
          mai.prc=FALSE,
          plot.extras = NA,
          source.plot=NA,
          image.size="800x1100",
          pointsize=12,
          res=NA,
          ps.paper="letter",
          ps.width=8,
          ps.height=11,
          returnVl=TRUE,
          saveFlag=FALSE,
          saveName="Splot.RData")
```

Arguments

`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

`plot.calls` character vector containing plot calls

`Iflag` Logical vector indicating if the plot in the layout is interactive

`figTypes` Character vector indicating the type of plot. Currently this argument is not needed, but will be useful for extensions that will be made in future versions.

`mai.mat` n x 4 matrix of values to be passed in for each plots par mai. n is equal to the length of plot.calls. If NA, uses default margins

`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 plot.calls. 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 required, an NA should be used

`source.plot` Indicates what image output the application should produce postscript, tiff, png, or jpeg. It can be a character vector of any combination of ps, png, jpeg, or tiff. If NA, the default will make a png file

`image.size` character indicating resize value of image,'width'x'height'

`pointsize` pointsize of image. passed into device call

`res` resolution of image, passed into device call if png or jpeg

`ps.paper` postscript paper argument if postscript is created

`ps.width` postscript width argument if postscript is created

`ps.height` postscript height argument if postscript is created

`returnVl` Should Splot object be returned

`saveFlag` Should Splot object be saved

`saveName` If saveFlag, path file name to save object

Details

This functio initializes a Splot object. This object stores information for constructing a layout of figures, as well as making any of thos figures interactive in a html webpage utilizing java tool-tip.

Users are encouraged to read the package vignette which includes a detailed discussion of all function arguments as well as several useful examples.

Value

If returnVl, an object of the class 'Splot'

Note

This function only sets up an SPlot object. It does not make the interactive figure.

Author(s)

Lori A. Shepherd, Dan P. Gaile

References

http://www.R-project.org

http://www.onlamp.com/pub/a/onlamp/2007/07/05/writing-advanced-javascript.html

http://www.walterzorn.com/tooltip/tooltip_e.htm

See Also

`sendplot`, `makeImap`, `makeSplot`, `layout`

Examples

```
# Please see vignette or makeSplot for example
```

makeCharacter *Converts Data Frame Entries Into Type Character*

Description

NOT CALLED BY USER. The makeCharacter function is utilized by makeSplot to convert data frame entries from type factor to type character for more efficient subsetting and writting to files.

Usage

```
makeCharacter(DF)
```

Arguments

`DF`	list containing data frames dat and dat2. This object is output from the makeScatterDF or makeImageDF function

Details

This function is a convenience function. It converts the data frames of a MapObj (output from `makeImap`) into character matricies. This speeds up the writing of data to the html file.

Value

List continaing character versions of data frames.

Note

NOT CALLED BY USER. INTERNAL HELPER FUNCTION

Author(s)

Lori A. Shepherd, Daniel P. Gaile

See Also

`makeSplot`

Examples

```
# not called by user
```

makeImageDF — *Maps R x and y point coordinates to pixil coordinates for tool-tip interactivity*

Description

NOT CALLED BY USER. The makeImageDF function is utilized by makeImap to map a set of R x and y coordinates to their corresponding pixil x and y coordinates. It also sets up data.frames of tool-tip information for display purpses

Usage

```
makeImageDF(Splot,
            xy.type,
            xlim, ylim,
            x.pos,y.pos,
            boundingPt,
            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)
```

Arguments

`Splot` An Object of the class Splot

`xy.type` Indication of how the xpos and ypos values should be treated.Current options are "image.midpoint","image.boundaries", or "image.box".See details or vignette.

`xlim` x limit of figure

`ylim` y limit of figure

`x.pos` numeric vector of x values for interactive points

`y.pos` numeric vector of y values for interactive points

`boundingPt` List with up.left and low.right pixil coordinates of the desired interactive figure's plotting region, as determined by `automapPts`

`x.labels` data frame of n x m which contains values relating to the xpos. n should be the length of the xpos argument if xy.type is "image.midpoints" and length of xpos - 1 if xy.type is "image.boundaries" or "image.box". m columns contains information regarding sample. This information is displayed in the interactive plot window

`y.labels` data frame of n x m which contains values relating to the yvec. n should be the length of the ypos argument if xy.type is "image.midpoints" and length of ypos - 1 if xy.type is "image.boundaries" or "image.box". m columns contains information regarding sample. This information is displayed in the interactive plot window

`xy.labels` list of matricies. All matricies should be of n x m where n is the length of y and m is the length of x when xy.type is "image.midpoint". All matricies should be of n x m where n is the length of y -1 and m is the length of x - 1 when xy.type is "image.boundaries" or "image.box". 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 xpos. n should be the length of the xpos argument if xy.type is "image.midpoints" and length of xpos - 1 if xy.type is "image.boundaries" or "image.box". m columns contains information regarding sample. 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 ypos. n should be the length of the ypos argument if xy.type is "image.midpoints" and length of ypos - 1 if xy.type is "image.boundaries" or "image.box". m columns contains information regarding sample. This information is displayed as hyperlinks in the interactive plot window.

`xy.links` list of matricies. All matricies should be of n x m where n is the length of y and m is the length of x when xy.type is "image.midpoint". All matricies should be of n x m where n is the length of y -1 and m is the length of x - 1 when xy.type is "image.boundaries" or "image.box". 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. May be a data.frame or matrix of n x m where n is the length of ypos and m is the length of xpos, a vector of length xpos indicating xpos specific links that will be repeated, a vector of length ypos indicating ypos specific links that will be repeated, a non NA value of length 1 that will be repeated for all points, or a vector of length xpos*ypos. This assumes xy.type is "image.midpoints". If xy.type is "image.boundaries" or "image.box", then the lengths should be xpos-1 and ypos - 1. instead of xpos and ypos

`x.images` data frame of n x m which contains paths to images relating to the xpos. n should be the length of the xpos argument if xy.type is "image.midpoints" and length of xpos - 1 if xy.type is "image.boundaries" or "image.box". m columns contains information regarding sample. 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 ypos. n should be the length of the ypos argument if xy.type is "image.midpoints" and length of ypos - 1 if xy.type is "image.boundaries" or "image.box". m columns contains information regarding sample. This information is displayed as images in the interactive plot window.

`xy.images` list of matricies. All matricies should be of n x m where n is the length of y and m is the length of x when xy.type is "image.midpoint". All matricies should be of n x m where n is the length of y -1 and m is the length of x - 1 when xy.type is "image.boundaries" or "image.box". This information is displayed in the interactive plot window as images. The values in these matricies should be complete paths to images

Details

This function uses the x and y limits of the R plots and the boundaries of the figure to convert plot points to pixil coordinates.

image.box will make rectangular regions for an image interactive based on the given xpos and ypos boundaries. image.midpoints and image.boundaries will make the center of the regions of an image interactive based on the given xpos and ypos. image.midpoints assumes the xpos and ypos are the actual locations. image.boundaries assumes the xpos and ypos are the boundary cuts and calculates the midpoints.

The data matricies are checked for proper lengths and returned as a MapObj.

Value

List containing objects with interactive information.

Note

NOT CALLED BY USER. INTERNAL HELPER FUNCTION

Author(s)

Lori A. Shepherd, Daniel P. Gaile

See Also

`makeImap`, `automapPts`

Examples

```
# not called by user
```

makeImap *Adds Interactive Information to 'Splot' Object*

Description

The makeImap function will add all information needed to make a given figure in a Splot Object interactive. This function may be run more than once for a given figure.

Usage

```
makeImap(Splot,
         figure=1,
         xy.type=NA,
         x.pos,
         y.pos,
         x.right.pos=NA,
         y.bottom.pos=NA,
         spot.radius = 5,
         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,
         sep.chr=":",
         font.type="Helvetica",
         font.color="black",
         font.size="12",
         bg.color="#D6E3F6",
         fname.root="Splot",
         dir="./",
         automap.method="mode",
         bb.clr=NA,
         bb.cex=2,
         returnVl=TRUE,
         saveFlag=FALSE,
         saveName="Splot.RData",
         cleanDir=TRUE)
```

Arguments

`Splot`	An Object of the class Splot

`figure` Indicates which plot figure data corresponds to; this matches the numeric indication in the layout matrix.

`xy.type` Indication of how the xpos and ypos values should be treated. Current options are "points", "image.midpoints", "image.boundaries", "image.box", "circle", "rect", and "poly". See details or vignette.

`x.pos` numeric vector of x values for interactive points. If xy.type is "rect", a numeric vector of the x.left position of rectangle[s].

`y.pos` numeric vector of y values for interactive points. If xy.type is "rect", a numeric vector of the y.top position of rectangle[s]

`x.right.pos` If xy.type is "rect", a numeric vector of the x.right position of rectangle[s]

`y.bottom.pos` If xy.type is "rect", a numeric vector of the y.bottom position of rectangle[s]

`spot.radius` radius of circle in pixels indicating area that will be interactive around the center of graphed points.used when xy.type is "points", "image.midpoints","image.boundaries", or "circle"

`x.labels` data frame of n x m which contains values relating to the xpos. This information is displayed in the interactive plot window. See details or vignette

`y.labels` data frame of n x m which contains values relating to the ypos. This information is displayed in the interactive plot window. See details or vignette

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

`x.links` data frame of n x m which contains web addresses for links relating to the xpos. This information is displayed as hyperlinks in the interactive plot window. See details or vignette

`y.links` data frame of n x m which contains web addresses for links relating to the ypos. This information is displayed as hyperlinks in the interactive plot window. See details or vignette

`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. See details or vignette

`asLinks` contains complete web address for points that should be treated as hyperlinks. See details or vignette

`x.images` data frame of n x m which contains paths to images relating to the xpos. n should be the length of the xpos argument if xy.type is "image.midpoints" and length of xpos - 1 if xy.type is "image.boundaries" or "image.box". m columns contains information regarding sample. 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 ypos. n should be the length of the ypos argument if xy.type is "image.midpoints" and length of ypos - 1 if xy.type is "image.boundaries" or "image.box". m columns contains information regarding sample. This information is displayed as images in the interactive plot window.

`xy.images` list of matricies. All matricies should be of n x m where n is the length of y and m is the length of x when xy.type is "image.midpoint". All matricies should be of n x m where n is the length of y -1 and m is the length of x - 1 when xy.type is "image.boundaries" or "image.box". This information is displayed in the interactive plot window as images. The values in these matricies should be complete paths to images

`sep.chr` seperation character in tool tip that distinguishes value from indicator. If a single value, duplicated for all. It may be a vector of values corresponding to each tooltip entry.

`font.type` font type for tool-tip. Currently support fonts are Arial, Helvetica,and sans-serif

`font.color` font color for tool-tip

`font.size` font size in tool-tip

`bg.color` background color of tool-tip

`fname.root` Base name to use for .png file name

`dir` directory path to where files should be created

`automap.method` Method to detect upper and lower bounds. Current options are "mode" or "median"

`bb.clr` vector of possible color choices for automatic detection bound point. see details or vignette

`bb.cex` Size of automatic detection bound point. see details or vignette

`returnVl` Should Splot object be returned

`saveFlag` Should Splot object be saved

`saveName` If saveFlag, path file name to save object

`cleanDir` logical indicating if the intermediate files generated by the application should be deleted. The files deleted are of no consequence to the user; they are needed to identify correct mapping

Details

Users are encouraged to see vignette for further details and several examples.

Two of the most common reasons for automap failing to detect bounding points concerns the color of 'phantom' boundings points with the color of the graphs, and the figure regions/resize value. If the 'phantom' point color is the same as the color used in that area of the graph, a difference will not be recorded. The point color is determined by bb.clr. By default, if bb.clr is left as NA, we try the following colors: blue, red, black, white, green. If the size of the plotting region is small, and therefore the point size is small, or if the resize value is very large, the difference of color for a single point will not register. The point size does not equate to a pixel. In this case the user may try increasing the size of the 'phantom' points added through the bb.cex argument.

xytype refers to how the x.pos and y.pos are treated. The possible options are "points", "image.midpoints", "image.boundaries", "image.box", "circle", "rect", and "poly". "Points","image.midpoints", and "circle" are handled the same. They assume the user is passing in x.pos and y.pos locations for a circle/point. image.midpoints is slightly different only in that it assumes the points refer to the center of regions in an image. The html image map will be a "circle". x.pos and y.pos will be the same length. "image.boundaries" assumes the x.pos and y.pos locations are referring to an image. The x.pos and y.pos locations are the boundaries of the regions, in other words, they are indications of x-axis and y-axis grid lines. The function will automatically calculate midpoints of the region and continues as if image.midpoints. "image.box" assumes x.pos and y.pos locations are referring to an image. The x.pos and y.pos locations are the boundaries of the regions,in other words, they are indications of x-axis and y-axis grid lines. The function calculates the boundaries of each rectangular image region to make interactive. The html image map will be a "rect". "rect" assumes the interactive regions are rectangular regions. x.pos refers to the left x coordinate[s] of the rectangular region[s]. y.pos refers to the top y coordinates[s] of the rectangular region[s]. When xy.type is "rect", x.right.pos and y.bottom.pos must also be specified indicating the right x and bottom y coordinate[s] respectively. x.pos, y.pos, x.right.pos, and y.bottom.pos will all be the same length. The html image map will be a "rect". "poly" assumes that one, and only one interactive polygon region is being added. The x.pos and y.pos therefore are the x and y vertices locations. The html image map will be a "poly".

The dimensions of x.lables, y.labels, xy.labels, x.links, y.links, xy.links, and asLinks will depend on xy.type. If xy.type is "points","circle",or "rect" x.labels, y.lables, x.links, and y.links will have the dimensions n by m where n is equal to the length of x.pos. asLinks will also be of length x.pos, or a single value that will be repeated. xy.labels and xy.links will be a list of matricies where each matrix is also n by m where n is equal to the length of x.pos. If xy.type is "image.midpoints" the following is true. x.labels and x.links should be n by m where n is equal to the length of x.pos, just as y.lables and y.links will be of length y.pos. xy.lables and xy.links will be lists of n by m matricies where n is the length of y.pos and m is the length of x.pos. asLinks can be of length x.pos, y.pos, (x.pos*y.pos), or 1. If xy.type is "image.boundaries" or "image.box" are very similar to "image.midpoints". Instead of length x.pos and y.pos the length is x.pos-1 and y.pos-1. If xy.type is "poly", x.lables, y.lables, x.links, y.links, and the matricies in xy.lables and xy.links are all 1 by m. asLinks is a single values.

Value

If returnVl, returns updated Splot object. Also creates .png and .tif files used for automatic detection

Note

Automatic detection is currently only functional for linux/unix users. A windows extension is being worked on for future version.

See details and vignette for notes on common reasons for automap failing.

Author(s)

Lori A.Shepherd, Daniel P. Gaile

References

Eric Kort (2006). rtiff: A tiff reader for R.. R package version 1.1.

See Also

`initSplot`,`sendplot`, `makeSplot`,`rtiff`

Examples

```
# Please see vignette or makeSplot for example
```

makePolyDF *Maps R x and y point coordinates to pixil coordinates for tool-tip interactivity*

Description

NOT CALLED BY USER. The makePolyDF function is utilized by makeImap to map a set of R x and y coordinates to their corresponding pixil x and y coordinates. It also sets up data.frames of tool-tip information for display purpses

Usage

```
makePolyDF(Splot,
           xlim, ylim,
           x.pos,
           y.pos,
           boundingPt,
           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)
```

Arguments

`Splot` An Object of the class Splot

`xlim` x limit of figure

`ylim` y limit of figure

`x.pos` numeric vector of x values for interactive points

`y.pos` numeric vector of y values for interactive points

`boundingPt` List with up.left and low.right pixil coordinates of the desired interactive figure's plotting region, as determined by `automapPts`

`x.labels` data frame of n x m which contains values relating to the x axis. n should be the length of the x.pos argument. m columns contains information regarding sample. 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. n should be the length of the y.pos argument. m columns contains information regarding sample. This information is displayed in the interactive plot window

`xy.labels`	list of matricies. All matricies should be of n x m where n is the length of x.pos and m is the length of y.pos. 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. n should be the length of the x.pos argument. m columns contains information regarding sample. 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. n should be the length of the y.pos argument. m columns contains information regarding sample. This information is displayed as hyperlinks in the interactive plot window.
`xy.links`	list of matricies. All matricies should be of n x m where n is the length of y.pos and m is the length of x.pos. 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 xpos. n should be the length of the xpos argument if xy.type is "image.midpoints" and length of xpos - 1 if xy.type is "image.boundaries" or "image.box". m columns contains information regarding sample. 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 ypos. n should be the length of the ypos argument if xy.type is "image.midpoints" and length of ypos - 1 if xy.type is "image.boundaries" or "image.box". m columns contains information regarding sample. This information is displayed as images in the interactive plot window.
`xy.images`	list of matricies. All matricies should be of n x m where n is the length of y and m is the length of x when xy.type is "image.midpoint". All matricies should be of n x m where n is the length of y -1 and m is the length of x - 1 when xy.type is "image.boundaries" or "image.box". This information is displayed in the interactive plot window as images. The values in these matricies should be complete paths to images

Details

This function uses the x and y limits of the R plots and the boundaries of the figure to convert plot points to pixil coordinates.

The function will make a polygon region using the coordinates in x.pos, y.pos. Only one polygon region can be added at a time.

The data matricies are checked for proper lengths and returned as a MapObj.

Value

List containing objects with interactive information.

Note

NOT CALLED BY USER. INTERNAL HELPER FUNCTION

Author(s)

Lori A. Shepherd, Daniel P. Gaile

See Also

makeImap,automapPts

Examples

```
# not called by user
```

---

makeRectDF *Maps R x and y point coordinates to pixil coordinates for tool-tip interactivity*

Description

NOT CALLED BY USER. The makeRectDF function is utilized by makeImap to map a set of R x and y coordinates to their corresponding pixil x and y coordinates. It also sets up data.frames of tool-tip information for display purpses

Usage

```
makeRectDF(Splot,
            xlim, ylim,
            x.left,y.top,
            x.right,y.bottom,
            boundingPt,
            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)
```

Arguments

`Splot` An Object of the class Splot

`xlim` x limit of figure

`ylim` y limit of figure

`x.left` numeric vector of left x values for interactive rectangles

`y.top` numeric vector of top y values for interactive rectangles

`x.right` numeric vector of right x values for interactive rectangles

`y.bottom` numeric vector of bottom y values for interactive rectangles

`boundingPt` List with up.left and low.right pixil coordinates of the desired interactive figure's plotting region, as determined by `automapPts`

`x.labels` data frame of n x m which contains values relating to the x axis. n should be the length of the x.left argument. m columns contains information regarding sample. 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. n should be the length of the y.top argument. m columns contains information regarding sample. This information is displayed in the interactive plot window

`xy.labels` list of matricies. All matricies should be of n x m where n is the length of x.left and m is the length of y.top. 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. n should be the length of the x.left argument. m columns contains information regarding sample. 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. n should be the length of the y.top argument. m columns contains information regarding sample. This information is displayed as hyperlinks in the interactive plot window.

`xy.links` list of matricies. All matricies should be of n x m where n is the length of y.top and m is the length of x.left. 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. Should be equal to the length of x.left

`x.images` data frame of n x m which contains paths to images relating to the xpos. n should be the length of the xpos argument if xy.type is "image.midpoints" and length of xpos - 1 if xy.type is "image.boundaries" or "image.box". m columns contains information regarding sample. 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 ypos. n should be the length of the ypos argument if xy.type is "image.midpoints" and length of ypos - 1 if xy.type is "image.boundaries" or "image.box". m columns contains information regarding sample. This information is displayed as images in the interactive plot window.

`xy.images` list of matricies. All matricies should be of n x m where n is the length of y and m is the length of x when xy.type is "image.midpoint". All matricies should be of n x m where n is the length of y -1 and m is the length of x - 1 when xy.type is "image.boundaries" or "image.box". This information is displayed in the interactive plot window as images. The values in these matricies should be complete paths to images

Details

This function uses the x and y limits of the R plots and the boundaries of the figure to convert plot points to pixil coordinates.

The function will make a rectangle region using the coordinates in x.left, y.top, x.right, y.bottom. The first rectangle would have the coordinates (x.left[1], y.bottom[1]),(x.left[1], y.top[1]), (x.right[1], y.top[1]), (x.right[1], y.bottom[1]). etc.

The data matricies are checked for proper lengths and returned as a MapObj.

Value

List containing objects with interactive information.

Note

NOT CALLED BY USER. INTERNAL HELPER FUNCTION

Author(s)

Lori A. Shepherd, Daniel P. Gaile

See Also

makeImap, automapPts

Examples

```
# not called by user
```

makeScatterDF *Maps R x and y point coordinates to pixil coordinates for tool-tip interactivity*

Description

NOT CALLED BY USER. The makeScatterDF function is utilized by makeImap to map a set of R x and y coordinates to their corresponding pixil x and y coordinates. It also sets up data.frames of tool-tip information for display purpses

Usage

```
makeScatterDF(Splot,
              xlim,
              ylim,
              x.pos,
              y.pos,
              boundingPt,
              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)
```

Arguments

`Splot` An Object of the class Splot

`xlim` x limit of figure

`ylim` y limit of figure

`x.pos` numeric vector of x values for interactive points

`y.pos` numeric vector of y values for interactive points

`boundingPt` List with up.left and low.right pixil coordinates of the desired interactive figure's plotting region, as determined by `automapPts`

`x.labels` data frame of n x m which contains values relating to the x.pos. n should be the length of the x.pos argument. m columns contains information regarding sample. This information is displayed in the interactive plot window

`y.labels` data frame of n x m which contains values relating to the y.pos. n should be the length of the y.pos argument. m columns contains information regarding sample. This information is displayed in the interactive plot window

`xy.labels` list of matricies. All matricies should be of n x m where n is the length of y.pos and m is the length of x.pos. 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.pos. n should be the length of the x.pos argument. m columns contains information regarding sample. 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.pos. n should be the length of the y.pos argument. m columns contains information regarding sample. This information is displayed as hyperlinks in the interactive plot window.

`xy.links` list of matricies. All matricies should be of n x m where n is the length of y.pos and m is the length of x.pos. 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. May be a data.frame or matrix of n x m where n is the length of y.pos and m is the length of x.pos, a vector of length x.pos indicating x.pos specific links that will be repeated, a vector of length y.pos indicating y.pos specific links that will be repeated, a non NA value of length 1 that will be repeated for all points, or a vector of length x.pos*y.pos

`x.images` data frame of n x m which contains paths to images relating to the xpos. n should be the length of the xpos argument if xy.type is "image.midpoints" and length of xpos - 1 if xy.type is "image.boundaries" or "image.box". m columns contains information regarding sample. 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 ypos. n should be the length of the ypos argument if xy.type is "image.midpoints" and length of ypos - 1 if xy.type is "image.boundaries" or "image.box". m columns contains information regarding sample. This information is displayed as images in the interactive plot window.

`xy.images` list of matricies. All matricies should be of n x m where n is the length of y and m is the length of x when xy.type is "image.midpoint". All matricies should be of n x m where n is the length of y -1 and m is the length of x - 1 when xy.type is "image.boundaries" or "image.box". This information is displayed in the interactive plot window as images. The values in these matricies should be complete paths to images

Details

This function uses the x and y limits of the R plots and the boundaries of the figure to convert plot points to pixil coordinates.

The data matricies are checked for proper lengths and returned as a MapObj.

Value

List containing objects with interactive information.

Note

NOT CALLED BY USER. INTERNAL HELPER FUNCTION

Author(s)

Lori A. Shepherd, Daniel P. Gaile

See Also

`makeImap`, `automapPts`

Examples

```
# not called by user
```

`makeSplot` *Makes Static or Interactive Files*

Description

This function acts on an Splot object to make a static (png, jpeg, or postscript) image and if desired an interactive html.

Usage

```
makeSplot(Splot,
          fname.root="Splot",
          dir="./",
          overwriteSourcePlot = NA,
          makeInteractive=TRUE,
          overrideInteractive=NA,
          Default=TRUE,
          header="v3",
          window.size = "800x1100",
          returnObj = TRUE)
```

Arguments

`Splot`	An Object of the class Splot
`fname.root`	Base name to use for all file created
`dir`	directory path to where files should be created
`overwriteSourcePlot`	character or character vector, should static image generated be postscript, png, jpeg, or tiff. If this is NA, it uses what is specified in Splot object.
`makeInteractive`	logical, should interactive html file be created. If FALSE, only static (ps, png, or jpeg) files is generated
`overrideInteractive`	logical of length equal to Splot\$nfig. Indicates which figures of the layout should be interactive; if NA uses what is specified in Splot object
`Default`	logical, If default tool-tip region is set should it be included in html
`header`	May either be v1,v2, or v3. This determines which tooltip header will be in the html file. Each version has different features or works well with different web browsers. see sp.header for details.
`window.size`	size of the html window. Only effective when header=v3
`returnObj`	Should Splot object be returned

Details

users are encouraged to read vignette for more details and several examples.

Value

Generates a static image (ps, png, or jpeg). If makeInteractive, an interactive .html file is generated If returnObj, Splot object is returned.

Note

The interactive html plot currently only works in web browsers that implement java script.

The code used to create the javascript embedded in html file is a modified version of the javascript code or from the open source tooltip library. see reference links

Author(s)

Lori A. Shepherd, Daniel P. Gaile

References

http://www.R-project.org

http://www.onlamp.com/pub/a/onlamp/2007/07/05/writing-advanced-javascript.html

http://www.walterzorn.com/tooltip/tooltip_e.htm

See Also

sendplot, initSplot, makeImap, heatmap.send, imagesend, xy.send

Examples

```
#
#   Please see vignette for full example
#

library(sendplot)
library("rtiff")

# set up layout matrix
mat = matrix(1,nrow=12, ncol=13)
mat[9:12,] = 2
mat[,9:13] = 3
mat[1:2,] = 4

# set margins
mai.mat = matrix(.5,ncol=4,nrow=4)

# set matrix to be used in image call
myX = c(-1,-10,1,10,-5,0)
```

```
plot.calls = c(
"boxplot(count ~ spray, data = InsectSprays,col='lightgray')",
"plot(1:3,1:3,col='blue',xlab='',ylab=''); points(1:2,2:3,col='red')",
"image(1:2,1:3, z=matrix(myX,ncol=3,nrow=2), xlab='',ylab='')",
"plot(cos, xlim=c(-pi,3*pi), n=1001,col ='blue',xlab='',ylab='')"
)


plt.extras=list(
figure1="rect(xleft=c(3,1),
ytop=c(25,5),xright=c(4,2),ybottom=c(20,0));title(main='A', cex=3)",
figure2="polygon(x=c(2,2.5,3,2.5), y=c(1,2.5,1,1.5));title(main='B',
cex=3)",
figure3 ="title(main='C', cex=3)",
figure4="title(main='D', cex=3)"
)


#set up temporary directory
direct = paste(tempdir(),"/",sep="")
direct

#
#
# make Splot object
#
#


# initialize Splot object
Splot = initSplot(mat, plot.calls, mai.mat=mai.mat,
                  plot.extras=plt.extras)

# first look/get plot output  -- makes static image
Splot = makeSplot(Splot, fname.root="exToy", makeInteractive=FALSE,
                  dir=direct, returnObj=TRUE)







#
#
# Add regions for interactive toop-tip display
#
#


# makes two rectangle regions interactive in figure 1
Splot = makeImap(Splot, figure=1, xy.type="rect",
                 x.pos=c(3,1), y.pos=c(25,5),
                 x.right.pos=c(4,2),  y.bottom.pos=c(20,0),
                 x.labels = as.data.frame(list(
                                label=c("rect1","rect2"),
                                info=c("im a link", "im a link"))),
                 asLinks = "http://www.buffalo.edu",
                 y.labels = c("d1",NA),
                 fname.root="exToy", dir=direct, spot.radius=10,
                 font.size=20, font.color="cyan", bg.color="black")

# makes one of the outlier points interactive in the figure 1
Splot = makeImap(Splot, figure=1, xy.type="circle", x.pos=3, y.pos=7,
                 x.labels = list(label="point", info="im a link"),
                 asLinks = "http://www.buffalo.edu", dir=direct,
                 y.labels = "7", fname.root="exToy", spot.radius=20,
                 font.type="arial", font.size="20",
                 font.color="hotpink", bg.color="blue")


# make polygon region interactive in figure 2
Splot = makeImap(Splot, figure=2, xy.type="polygon",
                 x.pos=c(2,2.5,3,2.5), y.pos=c(1,2.5,1,1.5),
                 x.labels = as.data.frame(list(label = "Polygon")),
                 asLinks="http://www.bioinformatics.buffalo.edu",
                 y.labels = as.data.frame(list(value="data1",
                                               info="Im alink")),
                 fname.root="exToy", dir=direct, bb.cex=5, font.size=30,
                 font.type="sans-serif", font.color="purple")
# make data set of three points interactive in figure 2
x.lbls = as.data.frame(list(labels=c("point1", "point2", "points3"),
                         info=c("im a link", "im a link", "im alink"),
                         numVl = c("num1", "num2", "num3"),
                         num = c(1,2,3)))
y.links = as.data.frame(list(
  hyp1=c("http://www.buffalo.edu",
         "http://www.bioinformatics.buffalo.edu,http://www.buffalo.edu",
         "http://www.bioinformatics.buffalo.edu"),
   hyp2=c(NA, "http://www.buffalo.edu",NA)
))
asLinks=c("http://www.buffalo.edu",
          "http://www.bioinformatics.buffalo.edu","http://www.buffalo.edu")
Splot = makeImap(Splot, figure=2, xy.type="points", x.pos=1:3, y.pos=1:3,
           x.labels = x.lbls,  y.links=y.links,asLinks=asLinks,
           fname.root="exToy",bb.cex=5, spot.radius=20,dir=direct,
           font.color="green",bg.color="", font.size="14")


# makes each box of image interactive in figure 3
x.lbls=as.data.frame(list(Xnuml = 1:2,
                          Xvl2 = c("mx1", "mx2")))
y.lbls=as.data.frame(list(ynum1 = 1:3,
                          Yvl2 = c("my1", "my2", "my3")))
xy.lbls=list(label = matrix(c("image.box1","image.box2",
                              ”image.box3”,”image.box4”,
                              ”image.box5”,”image.box6”),ncol=2),
             numMat1 = matrix(1:6, ncol=2),
             vlMat2 = matrix(c(”xy1”,”xy2”,”xy3”,”xy4”,”xy5”,”xy6”),
                             ncol=2))
x.links=as.data.frame(list(
   Xhyp1 = c(”http://www.buffalo.edu,http://bioinformatics.buffalo.edu”,
             NA)))
y.links=as.data.frame(list(Yhyp2 = c(NA, ”http//www.buffalo.edu”, NA)))
xy.links=list(XYhyp = matrix(c(NA, ”http://www.buffalo.edu”, NA,
                              ”http://buffalo.edu”, NA, NA), ncol=2))
Splot = makeImap(Splot, figure=3, xy.type=”image.box”,
                 x.pos= c(.5,1.5,2.5), y.pos=c(.5,1.5,2.5,3.5),
                 x.labels=x.lbls, y.labels = y.lbls, xy.labels=xy.lbls,
                 x.links=x.links, y.links=y.links, xy.links=xy.links,
                 fname.root=”exToy”, bb.cex=5, spot.radius=10,dir=direct)




#
# add Default tool-tip
#
Splot = addDefault(
        Splot,
        data=c(”This is default”, ”data2”),
        data.labels=c(”label”, ”d2”),
        links=c(”http://www.buffalo.edu,http://www.bioinformatics.buffalo.edu”,
                ”http://www.bioinformatics.buffalo.edu”),
        links.labels=c(”hyp1”, ”hyp2”),
        font.size=10, font.type=”arial”, bg.color=”gray”, font.color=”yellow”)




#
#
# Now that regions have been set as interactive
#    make interactive html
#
#
Splot = makeSplot(Splot, fname.root=”exToy”, dir=direct, returnObj=TRUE)





#
#
#  Now demonstarte removing interactive regions for figures
#  Note: to see effect must remake plot with the makeSplot
#  above
# removes outlier point interactive in figure 1
# since it was the second set added subset = 2
Splot = removeImap(Splot, figure=1, subset=2)


# remove default region toop-tip
Splot = removeImap(Splot, figure="Default")


# remove all interactive sets for figure 2
Splot = removeImap(Splot, figure=2)
```

`mapMethod` *Finds Point Coordinates*

Description

NOT CALLED BY USER. The mapMethod is utilized by the getBound function to find the up.left and low.right pixil coordinates of a given figures plotting bounding box

Usage

```
mapMethod(automap.method,
          temp)
```

Arguments

`automap.method`	Method to detect upper and lower bounds. Current options are "mode" or "median"
`temp`	binary matrix indicating where tif images differ, output from getBounds

Details

The mapMethod function takes output generated from the `getBounds` function. This output is a comparison of two pixmapRGB objects as a logical matrix, 0 if equal and 1 if different. The map method finds the areas of difference, calculating the upper left and lower right figure boundaries. These boundaries are used in converting R plot coordinates to pixil coordinates.

The boundary regions currently can be calculated by two techniques "mode" or "median". Median will take the central location of the regions found. Mode will take the median of the longest row and column to find the location of each region.

Value

A list with up.left and low.right bounding coordinates in pixils or NA if could not map correctly

Note

NOT CALLED BY USER. INTERNAL HELPER FUNCTION

Author(s)

Lori A.Shepherd, Daniel P. Gaile

See Also

`makeImap`, `automapPts`, `getBounds`

Examples

```
# not called by user
```

`removeImap` *Removes Interactive Information to 'Splot' Object*

Description

The removeImap function will remove interactive information for a given figure in a Splot Object. It will remove any information added by makeImap or addDefault.

Usage

```
removeImap(Splot,
           figure,
           subset=NA,
           returnVl=TRUE,
           saveFlag=FALSE,
           saveName="Splot.RData")
```

Arguments

`Splot`	An Object of the class Splot
`figure`	Indicates which plot figure to remove data from; this matches the numeric indication in the layout matrix. This may also be the value "Default" to remove a set Default tool-tip (see `addDefault`
`subset`	If default is a numeric indication of figure, which subsets of data to remove. If this is left as NA, all iMaps are removed.
`returnVl`	Should Splot object be returned
`saveFlag`	Should Splot object be saved
`saveName`	If saveFlag, path file name to save object

Details

The removeImap handles removing the Default toop-tip added by `addDefault` or any of a given figure's iMaps added by `makeImap`.

To remove all of a figure's iMaps, subset is NA. iMaps are stored in the order they are added. A subset will be a numeric indication of which iMap to remove based on this ordering.

Users are encouraged to see vignette for better details and examples.

Value

Returns or Saves the new Splot object with data removed

Note

Can only remove data from one figure at a time.

Author(s)

Lori A.Shepherd, Daniel P. Gaile

See Also

`addDefault`, `makeImap`, `initSplot`, `makeSplot`

Examples

```
# Please see vignette or makeSplot for example
```

---

sendimage *INTERACTIVE IMAGE - DEPRECATED*

Description

This function is a wrapper to sendplot that will create a single interactive image

Usage

```
sendimage(plot.call,
          x, y, z,
          z.value="value",
          x.lbls = NA,y.lbls=NA,xy.lbls=NA,
          x.links=NA, y.links=NA,
          xy.links=NA,asLinks=NA,
          mai=NA, mai.prc=FALSE,plt.extras=NA,
          bound.pt=FALSE, source.plot=NA,
          paint=FALSE, img.prog=NA,
          resize=”800x1100”,
          ps.paper=”letter”,ps.width=8,ps.height=11,
          fname.root=”test”,dir=”./”,header=”v2”,
          up.left=c(188,103),low.right=c(648,912),
          spot.radius=5, automap=FALSE, automap.method=”mode”)
```

Arguments

`plot.call` character vector containing single plot call

`x` vector of x locations for interactive points

`y` vector of y locations for interactive points

`z` vector of z values for image call

`z.value` character vector indicating the label for what the z argument holds.

`x.lbls` data frame of n x m which contains values relating to the x axis of the plot call. n should be the length of the x argument. m columns contains information regarding sample. This information is displayed in the interactive plot window

`y.lbls` data frame of n x m which contains values relating to the y axis of the plot. n should be the length of the y argument. m columns contains information regarding sample. This information is displayed in the interactive plot window

`xy.lbls` list of matricies. All matricies should be of n x m where n is the length of y and m is the length of x. 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 of the first plot. n should be the length of the x argument. m columns contains information regarding sample. 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 of the first plot. n should be the length of the y argument. m columns contains information regarding sample. This information is displayed as hyperlinks in the interactive plot window.

`xy.links` list of matricies. All matricies should be of n x m where n is the length of y and m is the length of x. 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. May be a data.frame or matrix of n x m where n is the length of y and m is the length of x, a vector of length x indicating x specific links that will be repeated, a vector of length y indicating y specific links that will be repeated, a non NA value of length 1 that will be repeated for all points, or a vector of length x*y

`mai` margins for each side of the plot.If NA uses default margins

`plt.extras` List of additional plotting calls that should be executed for the plot.

`mai.prc` logical indicating if mai mat values are percentages or hard coded values. If mai.proc is T, indicates percentage.

`bound.pt` logical indicating if red points should be plotted to aid in finding the upper left and lower right coordinates. If bound.pt is FALSE, indicates that up.left and low.right arguments are correct and will make the html file `source.plot` Indicates whether application should make a postscript file and then convert to png file, or if the png file should be made directly. This value is either ps, png, or NA. If NA the operating system is checked and the appropraite file format is output. Unix has a convert function that can convert a ps file to png file; we by default use this setup because we feel the postscript file maintains better quality. So on unix/linux systems if source.plot is NA, source.plot will be set to ps. Windows does not have this option, for this reason source.plot will be set to png if left NA

`paint` logical indicating if application should automatically open .png file for the user to view .png file and/or to retrieve needed bounding values of the plot call

`img.prog` If paint is TRUE, the command line call that will open a program to view .png file to retrieve pixil locations of interactive plot bounds. If this is left NA, the operating system is checked and a default program is used. For unix the default application is kolourpaint and for windows it is microsoft paint (mspaint)

`resize` character indicating resize value. The postscript version will be resized to this value when converted to .png.

`ps.paper` postscript paper argument

`ps.width` poscript width argument

`ps.height` postscript height argument

`fname.root` Base name to use for posctscript, .png, and html file names.

`dir` directory path to where files should be created

`header` May either be v1 or v2. This determines which tooltip header will be in the html file. Each version has different features or works well with different web browsers. see sp.header for details.

`up.left` The x and y value in pixels of the upper left hand corner of the plot call

`low.right` The x and y value in pixels of the lower right hand corner of the plot call.

`spot.radius` radius of circle in pixels indicating area that will be interactive around the center of interactive points

`automap` automatic detection of up.left and low.right bound points. Fully functional on linux/unix machines only.

`automap.method` Method to detect upper and lower bounds. Current options are mode or median

Details

This function is a wrapper for the sendplot function to create a single interacive image. See `sendplot` for more information.

Users are encouraged to read the package vignette which includes a detailed discussion of all function arguments as well as several useful examples.

Value

Creates a static .ps and .png file, and an interactive html file

Note

The interactive html plot currently only works in web browsers that implement java script.

The code used to create the javascript embedded in html file is a modified version of the javascript code or from the open source tooltip library. see reference links

This function is deprecated. Please see imagesend for updated version.

Author(s)

Lori A. Shepherd, Daniel P. Gaile

References

http://www.onlamp.com/pub/a/onlamp/2007/07/05/writing-advanced-javascript.html

http://www.walterzorn.com/tooltip/tooltip_e.htm

See Also

imagesend, initSplot, makeImap, makeSplot, sendplot

Examples

```
# load the library
   library("sendplot")

# set up vectors x,y, and z
   x = 1:4
   y = 1:8
   z = t(matrix(rnorm(32), ncol=4))

# create plot call for graph
  plot.calls = "image(x=x, y=y, z=z)"


#set up temporary directory
direct = paste(tempdir(),"/",sep="")
direct



# run sendImage
# note: we have already figured out appropriate up.left and low.right
#      values. if these were not known, the function should be run
#      with bound.pt=T (and maybe kolourpaint=T) to find pixil
#      locations

 sendimage(plot.call = plot.calls, x=x, y=y, z=z,
           up.left=c(100,99),low.right=c(738,917),
           bound.pt=FALSE, source.plot=NA, paint=FALSE,
           img.prog=NA,fname.root="testImg",dir=direct )
```

sendplot *CREATES WEB BROWSER INTERACTIVE PLOT - DEPRECATED*

Description

This function takes in a layout, a list of plot calls, and sample information. It generates a static image of plots. It also generates an html file with an interactive version of the image.

Usage

```
sendplot(mat, plot.calls,  x,y, mai.mat=NA, mai.prc=FALSE, xlim=NA, ylim=NA,
         z=NA,z.value="value", type="scatterplot", plt.extras =NA,
         x.lbls=NA, y.lbls=NA, xy.lbls=NA,
         x.links=NA, y.links=NA,
         xy.links=NA,asLinks=NA,
         bound.pt = FALSE,source.plot=NA,
         resize="800x1100", ps.paper="letter",ps.width=8,
         ps.height=11,fname.root="test",dir="./",header="v2",
         paint=FALSE, img.prog = NA,
         up.left=c(288,203),low.right=c(620,940),
         spot.radius=5, automap=FALSE, automap.method="mode")
```

Arguments

`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.
`plot.calls`	character vector containing plot calls
`mai.mat`	n x 4 matrix of values to be passed in for each plots par mai. n is equal to the length of plot.calls. If NA, uses default margins
`mai.prc`	logical indicating if mai mat values are percentages or hard coded values. If mai.proc is T, indicates percentage.
`xlim`	xlim values for the first plot call. This is required to set up intereactive plot for scatterplots. May be left NA. If this is NA and type is scatterplot the xlim will become the range of x values
`ylim`	ylim values for the first plot call. This is required to set up interactive plot for scatterplots.May be left NA. If this is NA and type is scatterplot the ylim will become the range of y values
`x`	vector of x values for the first plot call
`y`	vector of y values for the first plot call

z
vector of z values if the fist plot call is an image. If the plot call is not an image this may be left as NA

z.value
character vector indicating the label for what the z argument holds.

type
type of plot for the first plot call. Currently supports types are scatterplot or image

plt.extras
List of length equal to the number of plot.calls. 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 required, an NA should be used

x.lbls
data frame of n x m which contains values relating to the x axis of the first plot. n should be the length of the x argument. m columns contains information regarding sample. This information is displayed in the interactive plot window

y.lbls
data frame of n x m which contains values relating to the y axis of the first plot. n should be the length of the y argument. m columns contains information regarding sample. This information is displayed in the interactive plot window

xy.lbls
list of matricies. All matricies should be of n x m where n is the length of y and m is the length of x. 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 of the first plot. n should be the length of the x argument. m columns contains information regarding sample. 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 of the first plot. n should be the length of the y argument. m columns contains information regarding sample. This information is displayed as hyperlinks in the interactive plot window.

xy.links
list of matricies. All matricies should be of n x m where n is the length of y and m is the length of x. 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. May be a data.frame or matrix of n x m where n is the length of y and m is the length of x, a vector of length x indicating x specific links that will be repeated, a vector of length y indicating y specific links that will be repeated, a non NA value of length 1 that will be repeated for all points, or a vector of length x*y

bound.pt
logical indicating if red points should be plotted to aid in finding the upper left and lower right coordinates of a scatterplot. If bound.pt is FALSE, indicates that up.left and low.right arguments are correct and will make the html file

source.plot
Indicates whether application should make a postscript file and then convert to png file, or if the png file should be made directly. This value is either ps, png, or NA. If NA the operating system is checked and the appropraite file format is output. Unix has a convert function that can convert a ps file to png file; we by default use this setup because we feel the postscript file maintains better quality. So on unix/linux systems if source.plot is NA, source.plot will be set to ps. Windows does not have this option, for this reason source.plot will be set to png if left NA

`resize` character indicating resize value. The postscript version will be resized to this value when converted to .png.

`ps.paper` postscript paper argument

`ps.width` postscript width argument

`ps.height` postscript height argument

`fname.root` Base name to use for posctscript, .png, and html file names.

`dir` directory path to where files should be created

`paint` logical indicating if application should automatically open .png file for the user to view .png file and/or to retrieve needed bounding values of the first plot call. see details

`header` May either be v1 or v2. This determines which tooltip header will be in the html file. Each version has different features or works well with different web browsers. see sp.header for details.

`img.prog` If paint is TRUE, the command line call that will open a program to view .png file to retrieve pixil locations of interactive plot bounds. If this is left NA, the operating system is checked and a default program is used. For unix the default application is kolourpaint and for windows it is microsoft paint (mspaint)

`up.left` The x and y value in pixels of the upper left hand corner of the first plot call. see details

`low.right` The x and y value in pixels of the lower right hand corner of the first plot call. see details

`spot.radius` radius of circle in pixels indicating area that will be interactive around the center of graphed points

`automap` automatic detection of up.left and low.right bound points. Fully functional on linux/unix machines only.

`automap.method` Method to detect upper and lower bounds. Current options are mode or median

Details

The functions in the sendplot library allow R users to generate interactive plots with tool-tip content. A pair of files are created : a Portable Network Graphics (PNG) file which is a bitmap image and an HTML file which contains embedded Javascript code for dynamically generating tool-tips. When opened with a supported browser, the HTML file displays the PNG image and the user is able to mouse over and view tool-tip windows for user specified image locations. The information that appears in the tool-tip windows is user specified and highly customizable. The tool-tip functionality is provided by code from the wz_tooltip.js Javascript library (Zorn 2007) which is embedded in the HTML output.

The sendplot function constitutes the primary function of the sendplot library. It allows for the generation of interactive xy (i.e., scatter-plot) and image (i.e., heatmap) plots, which can contain any number of decorative (i.e., non-interactive) plots.

The creation of interactive plots with tool-tip content requires the development of the following components:

1. The static plot image. The library supports the following: a simple xy-plot (sendxy), a simple image plot (sendimage), a heatmap with decorative dendrograms (heatmap.send), or a flexible layout of plots which contains one interactive xy-plot or image plot (sendplot). The functions in the

sendplot library allow for the full complement of graphical bells and whistles which are available in R (e.g., custom axes, inclusion of legends, math symbols, etc.).

2. The plotted point to pixel mapping. The sendplot functions output an HTML file and a PNG image. The HTML file contains an image map which identifies the interactive regions of the PNG image (i.e., the regions for which a tool-tip will appear). The image map requires a mapping of the plotted point coordinates as specified in the R plotting calls that generated them to the corresponding pixel location on the final PNG image. The sendplot functions build this map by identifying the upper-left and lower-right locations in the original plotting coordinate system and in the final pixel coordinate system. The functions provide a convenient mechanism to accomplish this.

3. The tool-tip content lists. The sendplot functions allow users to specify x-specific, y-specific, and point specific (e.g., xy-specific) information to be displayed in the tool-tip.

The sendplot functions on windows machines are typically run in two iterations when creating interactive plots for the first time. In the first iteration, the PNG file is created and then opened in a program such as mspaint or kolourpaint so that the upper-left and lower-right pixel coordinates are identified. In the second iteration, the function is called again using the pixel coordinates identified in the first iteration and the PNG and HTML output files are created. Note: the first iteration need not be repeated for calls that use the sample plot type and output image size as the upper-left and lower-right pixel will not change.

On linux machines, there is an option for automatic detection of the upper-left and lower-right pixil coordinates. This utilizes ImageMagick's convert program install on most linux machines, and the rtiff R library's readTiff function. This eliminates the need for a second iteration. For windows users, this option is viable if the user has the ability to convert a PNG image to a TIF image; two iterations are still needed. See vignette for details.

Users are encouraged to read the package vignette which includes a detailed discussion of all function arguments as well as several useful examples.

Value

Creates a static .ps and .png file, and an interactive html file

Note

The x and y mappings to the interactive plot are created using the x and y vectors passed in as an argument to sendplot. Note: this could be handy if for example the user plotted more points to the first plot using the plt.extras argument. If the user wanted all points interactive, the x and y values of the sendplot argument would be a combination of all plotted points.

The interactive html plot currently only works in web browsers that implement java script.

The code used to create the javascript embedded in html file is a modified version of the javascript code or from the open source tooltip library. see reference links

This function is deprecated. Please see initSplot, makeImap, and makeSplot for updated version.

Author(s)

Daniel P. Gaile, Lori A. Shepherd

References

http://www.R-project.org

http://www.onlamp.com/pub/a/onlamp/2007/07/05/writing-advanced-javascript.html

http://www.walterzorn.com/tooltip/tooltip_e.htm

See Also

initSplot, makeImap, makeSplot, imagesend, xy.send, heatmap.send, layout

Examples

```
# see vignette for more advanced example


#
# first example of scatterplot
#
# note: we assume that the function has already been run once
#       to retrieve the pixel locations of the upper left and lower
#       right corners. If this had not been the case, the function would
#       need to have been run with bound.pt = T, perhaps paint = T
#
#
# The up.left and low.right values are correct if run in
#  unix/linux environment - for window users the coordinates will not be
#  correct


library(sendplot)

# create a layout with four plots
mat = matrix(c(rep(c(rep(3,8),rep(5,2)),1),
       rep(c(rep(1,8),rep(4,2)),14),
       rep(c(rep(2,8),rep(6,2)),2)),
       ncol=10,byrow=TRUE)

# create x and y points
x=rnorm(16)
y=rnorm(16)

# list of plot calls - what plot call to use for the four plots
plot.calls = c("plot(x,y,col='green', pch=3)","plot(0,0, col='purple',pch=22,bg='purple')","plot(1:3,1:3, type=

# create matrix of margin parameters
mai.mat = matrix(0, ncol=4, nrow=4, byrow=TRUE)
m1 = c(.25,0,.25,.5)
m2 = c(.4,0,.25,.5)
m3 = c(.1,0,.1,.5)
m4 = c(.25,0,.25,0)
mai.mat[1,] = m1
mai.mat[2,] = m2
mai.mat[3,] = m3
mai.mat[4,] = m4

# x and y limits of the first graph in the plot call
# in this case xlim and ylim values of plot(x,y,col='green', pch=3)
xlim = range(x, na.rm=TRUE)
xlim = c(xlim[1]-.1, xlim[2]+.1)
ylim = range(y, na.rm=TRUE)
ylim = c(ylim[1]-.1, ylim[2]+.1)

# data frame of sample information to display in interactive plot
# since scatterplot has equal number of pts no need to be x, y, or xy specific
x.lbls = list()
x.lbls$test = rep(c("a","b","c","d"),4)
x.lbls$num = 1:16
x.lbls = as.data.frame(x.lbls)



#set up temporary directory
direct = paste(tempdir(),"/",sep="")
direct


sendplot(mat, plot.calls, mai.mat,
         xlim=xlim, ylim=ylim,type="scatterplot",
         x=x,y=y,z=NA,
         x.lbls=x.lbls, y.lbls=NA, xy.lbls=NA, source.plot=NA,
         resize="1200x1700", fname.root="testScatterplot",dir=direct,
         paint=FALSE, bound.pt=FALSE,img.prog = NA,
         spot.radius=5, up.left=c(186,199),low.right=c(767,1264))

# there will now be a static postscript and .png file that may be viewed
# as well as an html file that can be opened with firefox that has
# interactive version



#
#
# second example of image
#
# note: we assume that the function has already been run once
#       to retrieve the pixel locations of the upper left and lower
#       right corners. If this had not been the case, the function would
#       need to have been run with perhaps paint = T
#
#
# The up.left and low.right values are correct if run in
#  unix/linux environment - for window users the coordinates will not be
#  correct
```

```
library(sendplot)

# create a layout with four plots
mat = matrix(c(rep(c(rep(3,8),rep(5,2)),1),
       rep(c(rep(1,8),rep(4,2)),14),
       rep(c(rep(2,8),rep(6,2)),2)),
       ncol=10,byrow=TRUE)

# create x and y points, and z matrix of values for image
y=c(1:5,10,20,22,30,36)
x=c(1,2,4,5,8)
z=matrix(rnorm(50), nrow=5, ncol=10)

# list of plot calls - what plot call to use for the four plots
plot.calls = c(”image(x=x, y=y, z=z)”,
   ”plot(0,0, col='purple',pch=22,bg='purple')”,”plot(1:3,1:3, type='b',pch=21,bg='red',col='red')”,”curve(x^3-

# create matrix of margin parameters
mai.mat = matrix(0, ncol=4, nrow=4, byrow=TRUE)
m1 = c(.25,0,.25,.5)
m2 = c(.4,0,.25,.5)
m3 = c(.1,0,.1,.5)
m4 = c(.25,0,.25,0)
mai.mat[1,] = m1
mai.mat[2,] = m2
mai.mat[3,] = m3
mai.mat[4,] = m4

# mock data frames of x specific and y specific data
x.lbls = list()
x.lbls$test = c(”a”,”b”,”c”,”d”,”e”)
x.lbls$num = 1:5
x.lbls = as.data.frame(x.lbls)
y.lbls = list()
y.lbls$test2 = rep(c(”f”,”g”,”h”,”i”,”j”),2)
y.lbls$num2 = 10:1
y.lbls=as.data.frame(y.lbls)

# mock list of data frames of xy specific data
xy.lbls = list()
xy.lbls$one = matrix(1,nrow=10,ncol=5)
xy.lbls$two = matrix(2,nrow=10,ncol=5)
xy.lbls$aa = matrix(”a”, nrow=10,ncol=5)




sendplot(mat, plot.calls, mai.mat,
        xlim=NA, ylim=NA,type=”image”,
        x=x,y=y,z=z, z.value=”value”,
        x.lbls=x.lbls, y.lbls = y.lbls, xy.lbls=xy.lbls,
        resize=”1200x1700”, fname.root=”testimage”,source.plot=NA,
          dir=direct, paint=FALSE,img.prog = NA, bound.pt=FALSE, spot.radius=8,
          up.left=c(163,156),low.right=c(790,1310))


# there will now be a static postscript and .png file that may be viewed
# as well as an html file that can be opened with firefox that has
# interactive version
```

sendxy *INTERACTIVE SCATTERPLOT- DEPRECATED*

Description

This function is a wrapper to sendplot that will create a single interactive scatterplot

Usage

```
sendxy(plot.call,
       x, y,
       xy.lbls = NA, x.lbls = NA,y.lbls=NA,
       x.links=NA, y.links=NA,
       xy.links=NA,asLinks=NA,
       xlim = NA, ylim = NA,
       mai=NA, mai.prc=FALSE,plt.extras=NA,
       bound.pt=FALSE, source.plot=NA,
       paint=FALSE,img.prog = NA,
       resize="800x1100",
       ps.paper="letter",ps.width=8,ps.height=11,
       fname.root="test",dir="./",header="v2",
       up.left=c(205,131),low.right=c(633,883),
       spot.radius=5, automap=FALSE, automap.method="mode")
```

Arguments

`plot.call`	character vector containing single plot call
`x`	vector of x locations for interactive points
`y`	vector of y locations for interactive points
`xy.lbls`	data frame of n x m which contatins values relating to the x axis of the plot call. n should be the length of the x argument. m columns contains information regarding sample. This information is displayed in the interactive plot window

`x.lbls` data frame of n x m which contains values relating to the x axis of the plot call. n should be the length of the x argument. m columns contains information regarding sample. This information is displayed in the interactive plot window

`y.lbls` data frame of n x m which contains values relating to the y axis of the plot. n should be the length of the y argument. m columns contains information regarding sample. 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 of the first plot. n should be the length of the x argument. m columns contains information regarding sample. 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 of the first plot. n should be the length of the y argument. m columns contains information regarding sample. This information is displayed as hyperlinks in the interactive plot window.

`xy.links` list of matricies. All matricies should be of n x m where n is the length of y and m is the length of x. 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. May be a data.frame or matrix of n x m where n is the length of y and m is the length of x, a vector of length x indicating x specific links that will be repeated, a vector of length y indicating y specific links that will be repeated, a non NA value of length 1 that will be repeated for all points, or a vector of length x*y

`xlim` xlim values for the plot. If left as NA, the range of x will be used.

`ylim` ylim values for the plot. If left as NA, the range of y will be used.

`mai` margins for each side of the plot.If NA uses default margins

`mai.prc` logical indicating if mai mat values are percentages or hard coded values. If mai.proc is T, indicates percentage.

`plt.extras` List of additional plotting calls that should be executed for the plot.

`bound.pt` logical indicating if red points should be plotted to aid in finding the upper left and lower right coordinates. If bound.pt is FALSE, indicates that up.left and low.right arguments are correct and will make the html file

`source.plot` Indicates whether application should make a postscript file and then convert to png file, or if the png file should be made directly. This value is either ps, png, or NA. If NA the operating system is checked and the appropraite file format is output. Unix has a convert function that can convert a ps file to png file; we by default use this setup because we feel the postscript file maintains better quality. So on unix/linux systems if source.plot is NA, source.plot will be set to ps. Windows does not have this option, for this reason source.plot will be set to png if left NA

`paint` logical indicating if application should automatically open .png file for the user to view .png file and/or to retrieve needed bounding values of the plot call.

`img.prog` If paint is TRUE, the command line call that will open a program to view .png file to retrieve pixil locations of interactive plot bounds. If this is left NA, the operating system is checked and a default program is used. For unix the default application is kolourpaint and for windows it is microsoft paint (mspaint) `resize` character indicating resize value. The postscript version will be resized to this value when converted to .png.

`ps.paper` postscript paper argument

`ps.width` poscript width argument

`ps.height` postscript height argument

`fname.root` Base name to use for posctscript, .png, and html file names.

`dir` directory path to where files should be created

`up.left` The x and y value in pixels of the upper left hand corner of the plot call

`header` May either be v1 or v2. This determines which tooltip header will be in the html file. Each version has different features or works well with different web browsers. see sp.header for details.

`low.right` The x and y value in pixels of the lower right hand corner of the plot call.

`spot.radius` radius of circle in pixels indicating area that will be interactive around the center of interactive points

`automap` automatic detection of up.left and low.right bound points. Fully functional on linux/unix machines only.

`automap.method` Method to detect upper and lower bounds. Current options are mode or median

Details

This function is a wrapper for the sendplot function to create a single interacive scatterplot. See `sendplot` for more information.

Users are encouraged to read the package vignette which includes a detailed discussion of all function arguments as well as several useful examples.

Value

Creates a static .ps and .png file, and an interactive html file

Note

The interactive html plot currently only works in web browsers that implement java script.

The code used to create the javascript embedded in html file is a modified version of the javascript code or from the open source tooltip library. see reference links

This function is deprecated. Please see xy.send for updated version.

Author(s)

Lori A. Shepherd, Daniel P. Gaile

References

http://www.onlamp.com/pub/a/onlamp/2007/07/05/writing-advanced-javascript.html

http://www.walterzorn.com/tooltip/tooltip_e.htm

See Also

initSplot, makeImap, makeSplot, xy.send, imagesend, heatmap.send

Examples

```
# load the library
  library("sendplot")

# create some vectors of points
   x1 = 1:7
   y1 = 1:7
   x2 = 7:1
   y2 = rep(4,7)
# create master vector of all points
   x = c(x1,x2)
   y = c(y1,y2)

# create data frame of information to display in interactive
   xy.lbls = list()
   xy.lbls$test = rep(c("a","b","c","d","e","f","g"),2)
   xy.lbls$num = 1:14
   xy.lbls = as.data.frame(xy.lbls)

# create plot.call for graph
   plot.calls = "plot(x1,y1,col='green', pch=3, cex=1.5); points(x2,y2,
pch=4, cex=1.5, col='purple')"


#
# Note this plot call could have also been run with
#
#   plot.calls = "plot(x1,y1,col='green', pch=3, cex=1.5)"
#
#  and then setting the
#        plt.extras$plot1 = "points(x2,y2,pch=4, cex=1.5, col='purple')"


#set up temporary directory
direct = paste(tempdir(),"/",sep="")
direct




# run sendxy
# note: we have already figured out appropriate up.left and low.right
#      values. if these were not known, the function should be run
#      with bound.pt=T (and maybe paint=T) to find pixil
#      locations
#
# The up.left and low.right values are correct if run in
#  unix/linux environment - for window users the coordinates will not be
# correct

  sendxy(plot.call = plot.calls, x=x, y=y,
         xy.lbls=xy.lbls, plt.extras=NA,
         bound.pt=FALSE, source.plot=NA, paint=FALSE,
         img.prog=NA,fname.root="testXY",dir=direct,
         up.left=c(124,130),low.right=c(713,883))
```

spheader *HEADER INFORMATION FOR HTML FILE*

Description

Stores header information, tooltip information, for html file

Format

Character vector containing lines of html header information

Details

This file contains header information for an html file. It also contains javascript tooltip necessary for interactive plot. The different versions have different features or work on different web browsers. v1 works well with firefox and displays information in the upper right corner of the web browser v2 works well with firefox and displays information at the mouse location. v3 is the same as v2 except it allows control of the html window size. The default window size is 800x1100. To change window size, alter the window.size argument in `makeSplot`, `imagesend`, `xy.send`, or `heatmap.send`.

Note

This dataset is used within the sendplot function. There is no need for the user to ever call this dataset.

Source

http://www.walterzorn.com/tooltip/tooltip_e.htm

References

http://www.onlamp.com/pub/a/onlamp/2007/07/05/writing-advanced-javascript.html

`Splot-class` *Splot 'sendplot' Object*

Description

The Splot object contains all data needed to generate a static layout of plots, as well as any data mappings for a html file to allow for java tool-tip display.

Author(s)

Lori A. Shepherd, Daniel P. Gaile

See Also

`initSplot`, `makeImap`

`writeArea` *Writes Data Frame Entries to File In Interactive Format*

Description

NOT CALLED BY USER. These functions are utilized by the writeToHTML functions to access entries of a data frame and convert entries into a format for interactive display. Output is written to HTML file.

Usage

```
writeCircle.1(DFs, cdat, ndat, obj)
writeCircle.2(DFs, cdat, ndat, obj)
writeRect.1(DFs, cdat, ndat)
writeRect.2(DFs, cdat, ndat, obj)
writePoly.1(DFs, cdat, ndat, obj)
writePoly.2(DFs, cdat, ndat, obj)
```

Arguments

`DFs`	output from makeCharacter function
`cdat`	Data frame containing data
`ndat`	vector containing names/identifiers for cdat
`obj`	list containing data frames dat and dat2. This object is output from the makeScatterDF or makeImageDF function

Details

The writeArea functions are utilized by the writeToHTML functions when an interactive figure is desired. It takes properly formated data and based on what imagemap region (i.e. circle, rect, poly, or default), writes data to an HTML file.

function.1 is utilized when using header = "v1" function.2 is utilized when using header = "v2" or "v3" The data is displayed in different ways depending on the header.

users are encouraged to see vignette for better descriptions and examples

Value

Line by Line output written to HTML file

Note

NOT CALLED BY USER. INTERNAL HELPER FUNCTION

The code used to create the javascript embedded in html file is a modified version of the javascript code or from the open source tooltip library. see reference links

Author(s)

Lori A. Shepherd, Daniel P. Gaile

References

http://www.onlamp.com/pub/a/onlamp/2007/07/05/writing-advanced-javascript.html

http://www.walterzorn.com/tooltip/tooltip_e.htm

See Also

`makeSplot`, `writeToHTML`, `sp.header`

Examples

```
# not called by user
```

`writeDefault` *Writes Default Tool-tip Region to imagemap of HTML*

Description

NOT CALLED BY USER. If a default toop-tip region has been set using `addDefault`, the information is converted to proper format and wrote to HTML file

Usage

```
writeDefault1(Splot)
writeDefault2(Splot)
```

Arguments

`Splot`	An Object of the class Splot

Details

If a default toop-tip region has been set using `addDefault`, the information is converted to proper format and wrote to HTML file

writeDefault1 is utilized when using header = "v1" writeDefault2 is utilized when using header = "v2" or "v3" The data is displayed in different ways depending on the header.

users are encouraged to see vignette for better descriptions and examples

Value

Line for default tool-tip region added to imagemap section of HTML

Note

NOT CALLED BY USER. INTERNAL HELPER FUNCTION

The code used to create the javascript embedded in html file is a modified version of the javascript code or from the open source tooltip library. see reference links

Author(s)

Lori A. Shepherd, Daniel P. Gaile

References

http://www.onlamp.com/pub/a/onlamp/2007/07/05/writing-advanced-javascript.html

http://www.walterzorn.com/tooltip/tooltip_e.htm

See Also

`makeSplot`, `addDefault`, `sp.header`

Examples

```
# not called by user
```

writeToHTML *Writes Data Frame Entries to File In Interactive Format*

Description

NOT CALLED BY USER. The writeToHTML functions are utilized by makeSplot to access entries of a data frame and convert entries into a format for interactive display.

Usage

```
writeToHTML1(obj,DFs, iType)

writeToHTML2(obj,DFs, iType)
```

Arguments

obj	list containing data frames dat and dat2. This object is output from the makeScatterDF or makeImageDF function
DFs	output from makeCharacter function
iType	indication of what type of image map region. Currently supported types are circle, rect, poly, or default

Details

The writeToHTML functions are utilized by the makeSplot when an interactive figure is desired. It takes properly formated data and based on what imagemap region (i.e. circle, rect, poly, or default), writes data to an HTML file.

HTML1 is utilized when using header = "v1" HTML2 is utilized when using header = "v2" or "v3" The data is displayed in different ways depending on the header.

users are encouraged to see vignette for better descriptions and examples

Value

Line by Line output

Note

NOT CALLED BY USER. INTERNAL HELPER FUNCTION

The code used to create the javascript embedded in html file is a modified version of the javascript code or from the open source tooltip library. see reference links

Author(s)

Lori A. Shepherd, Daniel P. Gaile

References

http://www.onlamp.com/pub/a/onlamp/2007/07/05/writing-advanced-javascript.html

http://www.walterzorn.com/tooltip/tooltip_e.htm

See Also

`makeSplot`, `writeArea`, `sp.header`

Examples

```
# not called by user
```

`xy.send` *INTERACTIVE SCATTERPLOT*

Description

This function is a wrapper to sendplot that will create a single interactive scatterplot

Usage

```
xy.send(plot.call,
        x.pos,
        y.pos,
        plot.extras = NA,
        mai.mat=NA, mai.prc=FALSE,
        xy.labels=NA,
        image.size="800x1100",
        spot.radius = 5,
        fname.root="Splot",
        dir="./",
        window.size = "800x1100",
        ...)
```

Arguments

`plot.call`	character vector containing single plot call
`x.pos`	vector of x locations for interactive points
`y.pos`	vector of y locations for interactive points
`plot.extras`	List of additional plotting calls that should be executed for the plot.
`mai.mat`	1 x 4 matrix of values to be passed in for each plots par mai. n is equal to the length of plot.calls. If NA, uses default margins.

`mai.prc` logical indicating if mai mat values are percentages or hard coded values. If mai.proc is T, indicates percentage.

`xy.labels` list of matricies. All matricies should be of n x m where n is the length of xpos or ypos. This information is displayed in the interactive plot window

`image.size` character indicating size of device.

`spot.radius` radius of circle in pixels indicating area that will be interactive around the center of graphed points

`fname.root` Base name to use for all files created.

`dir` directory path to where files should be created. Default creates files in working directory

`window.size` size of the html window

`...` additional arguments to the makeImap function

Details

This function is a wrapper for the sendplot function to create a single interacive scatterplot. See `initSplot`, `makeImap`, and `makeSplot` for more information.

Users are encouraged to read the package vignette which includes a detailed discussion of all function arguments as well as several useful examples.

Value

Creates a static and interactive scatterplot

Note

The interactive html plot currently only works in web browsers that implement java script.

The code used to create the javascript embedded in html file is a modified version of the javascript code or from the open source tooltip library. see reference links

Author(s)

Lori A. Shepherd, Daniel P. Gaile

References

http://www.onlamp.com/pub/a/onlamp/2007/07/05/writing-advanced-javascript.html

http://www.walterzorn.com/tooltip/tooltip_e.htm

See Also

`initSplot`, `makeImap`, `makeSplot`, `imagesend`, `heatmap.send`, `sendplot-package`, `sendxy`

Examples

```
library(sendplot)
library(rtiff)


plot.call=c("plot(mtcars$hp,mtcars$mpg,xlab='gross horsepower',
                   ylab='miles per gallon',axes=FALSE,pch=mtcars$cyl,
                   col=mtcars$am+1,cex=0.875,
                   main='Motor Trend Car Road Tests')")

plot.extras=c("axis(1);axis(2);
              legend(200,25,pch=rep(c(4,6,8),2),col=c(rep(1,3),rep(2,3)),
                     legend=paste(rep(c(4,6,8),2),'cylinders,',
                             c('automatic','manual')[c(rep(1,3),rep(2,3))]),cex=0.875)")


#set up temporary directory
direct = paste(tempdir(),"/",sep="")
direct


xy.send(plot.call=plot.call,
       y.pos=mtcars$mpg,x.pos=mtcars$hp,
       xy.labels = data.frame(name=rownames(mtcars),mtcars=mtcars),
       plot.extras=plot.extras,
       image.size="800x600",
       fname.root="exPlotXY", dir = direct, font.size=18)
```

Index