

# Package ‘somplot’

February 20, 2015

**Type** Package

**Title** Visualisation of hexagonal Kohonen maps

**Version** 1.6.4

**Date** 2013-07-21

**Author** Benjamin Schulz, Andreas Dominik

**Maintainer** Andreas Dominik <andreas.dominik@mni.thm.de>

**Depends** hexbin

**Description** The package provides the plot function som.plot() to create high quality visualisations of hexagonal Kohonen maps (self-organising maps).

**License** GPL-2

**LazyLoad** yes

**Repository** CRAN

**Date/Publication** 2013-07-22 17:19:25

**NeedsCompilation** no

## R topics documented:

somplot-package . . . . .	2
hexbinpie . . . . .	2
makehexbinplot . . . . .	4
som.plot . . . . .	5

<b>Index</b>	<b>8</b>
--------------	----------

somplot-package

*Plot hexagonal Kohonen maps*

---

**Description**

The package provides a function `som.plot()` for generating high-quality plots of hexagonal self-organising maps (SOMs), which have been created with Teuvo Kohonen's SOM\_PAK software.

**Details**

Package: somplot  
Type: Package  
Version: 1.6.1  
Date: 2013-01-30  
License: GPL  
LazyLoad: yes

Two files are necessary for plotting the result of a SOM\_PAK run:

- The output file of `visual` provides the mapping of input patterns to the codebook vectors.
- The input file holds all patterns of the dataset with class information in the last column.

A visualisation with default parameters can be created by calling the function `som.plot( visfile, datfile)`. The plot can be customised, by providing additional arguments for colouring, scaling, orientation or legend.

**Author(s)**

Benjamin Schulz, Andreas Dominik Maintainer: <andreas.dominik@mni.fh-giessen.de>

**References**

Schulz, B., Mauthe, T., Dominik, A., 2010. Visualisation of Kohonen Maps using R. THM University of Applied Sciences, Giessen, Germany Website, [Online]. Available at: <http://www.life-science-it.org/pages/research/projectSomplot.html> [Accessed 19 May 2011]. and <http://www.bioconductor.org/packages/2.5/bioc/html/hexbin.html>

---

hexbinpie*Function, used by som.plot to create plots of Kohonen maps*

---

**Description**

The function is used by `som.plot`. It is not necessary to call `hexbinpie` directly. The function draws pie charts in a hexagonal grid.

**Usage**

```
hexbinpie(x, y, kat, xbnds = range(x), ybnds = range(y),  
          hbc = NA, pal = NA, hex = "gray", circ = "gray50",  
          cnt = "black", show.counter.border, ...)
```

**Arguments**

<code>x</code>	vector of X coordinates
<code>y</code>	vector of Y coordinates
<code>kat</code>	vector of categories for each data point
<code>xbnds</code>	limits in X direction
<code>ybnds</code>	limits in Y direction
<code>hbc</code>	data frame holding the neurons
<code>pal</code>	colours to be used to plot the classes of input data
<code>hex</code>	colour for hexagons
<code>circ</code>	colour for circles
<code>cnt</code>	colour for labels in the pies
<code>show.counter.border</code>	percentile as limit for the display of labels in the pie charts.
<code>...</code>	more arguments for customising the plot

**Value**

The function returns no value

**Warning**

The function is called by `som.plot()` and not intended to be used directly.

**Author(s)**

Benjamin Schulz and Andreas Dominik

**References**

see function `som.plot()`

**Examples**

```
## Not run:  
hexbinpie(data$x, data$y, kat=data$kat, hbc = hbc, pal = pal, ...)  
  
## End(Not run)
```

---

makehexbinplot      *Function, used by som.plot to create plots of Kohonen maps*

---

### Description

The function is used by som.plot. It is not necessary to call makehexbinplot directly.

### Usage

```
makehexbinplot(data, col = NA, show.legend = TRUE, legend.width = 4,
  turn = FALSE, window.width = NA, window.height = NA, onlyDefCols = FALSE,
  scaleX = NA, scaleY = NA, scale = NA, new.xdim = NA, new.ydim = NA,
  show.box = TRUE, show.axis = FALSE, edit.cols = FALSE,
  show.counter.border = 0.98, ...)
```

### Arguments

data	data frame to be plotted
col	default colours for the classes of the dataset. Possible values include: <ul style="list-style-type: none"> <li>• default value: NA. Colours are generated by rainbow()</li> <li>• vector of colour definitions</li> <li>• data frame with name of a class in column 1 and colour definitions in column 2. If the number of defined colours is smaller than the number of classes in the dataset, colours for the remaining classes are generated by rainbow</li> </ul>
show.legend	default: TRUE; defines if colour legend is displayed
legend.width	default: 4; Width of legend
turn	default: FALSE; swap X and Y axis
window.width	default: NA; width of the window
window.height	default: NA; height of the window
onlyDefCols	default: FALSE; if TRUE, all undefined colours are replaced by white
scaleX	default: NA; scale factor for X axis
scaleY	default: NA; scale factor for Y axis
scale	default: NA; scale factor for X and Y axis
new.xdim	default: NA; scale X axis to specified number of neurons
new.ydim	default: NA; scale Y axis to specified number of neurons
show.box	default: TRUE; show frame around the plot
show.axis	default: FALSE; show x and Y axis
edit.cols	default: FALSE; if TRUE, a dialog box opens and allows editing of all color definitions
show.counter.border	percentile as limit for the display of labels in the pie charts.
...	In addition all arguments accepted by plot() are allowed.

**Value**

The function does not returns a value.

**Warning**

The function is called by som.plot() and not intended to be used directly.

**Author(s)**

Benjamin Schulz, Andreas Dominik

**References**

see function som.plot()

**Examples**

```
## Not run:  
(data.frame(coo[, c(1,2)], kat = dat[-1, dat[1,1]+1]), ...)  
  
## End(Not run)
```

---

som.plot

*Function to plot hexagonal Kohonen maps*

---

**Description**

The function plots hexagonal self-organising maps (SOMs), which have been created with Teuvo Kohonen's SOM\_PAK software.

The plot shows an hexagonal lattice representing the codebook vectors of the SOM. Inside each hexagon a pie chart is drawn, showing the relative percentage of pattern classes, mapped to this codebook vector. The size of each pie chart is adjusted according to the number of mapped patterns.

Several arguments allow customisation of size, scale and colouring of the plot.

The function needs two files of the SOM\_PAK file set:

- The output of visual provides the mapping of input patterns to the codebook vectors.
- The input file holding the mapped dataset with class information in the last column.

**Usage**

```
som.plot(visfile, datfile, ...)
```

**Arguments**

visfile	Name of the output file of visual. visual is the tool of SOM_PAK which maps pattern to the codebook vectors of a self-organising map.\ The function reads files in the format created by visual.
datfile	Name of the SOM_PAK input file. This files contains all input patterns and class information for each pattern. som-plot only uses the last colums of the file; i.e. the class information for each pattern.\ The function reads files in the format that is used by SOM_PAK.
...	<p>Many other arguments can be given to control the appearence of the plot:</p> <p>‘col’, default colours for the classes of the dataset. Possible values include:  ‘default value: NA.’ Colours are generated by rainbow()  ‘vector’ of colour definitions  ‘data frame’ with name of a class in column 1 and colour definitions in column 2.</p> <p>If the number of defined colours is smaller then the number of classes in the dataset, colours for the remaining classes are generated by rainbow.</p> <p>‘onlyDefCols’ default: FALSE; if TRUE, all undefined colours are replaced by white</p> <p>‘edit.cols’ default: FALSE; if TRUE, a dialog box opens and allows editing of all color definitions</p> <p>‘show.legend’ default: TRUE; defines if colour legend is displayed</p> <p>‘legend.width’ default: 4; Width of legend</p> <p>‘turn’ default: FALSE; swap X and Y axis</p> <p>‘window.width’ default: NA; width of the window</p> <p>‘window.height’ default: NA; height of the window</p> <p>‘show.box’ default: TRUE; show frame around the plot</p> <p>‘show.axis’ default: FALSE; show x and Y axis</p> <p>‘scaleX’ default: NA; scale factor for X axis (overwrites new.xdim argument)</p> <p>‘scaleY’ default: NA; scale factor for Y axis (overwrites new.ydim argument)</p> <p>‘scale’ default: NA; scale factor for X and Y axis (overwrites other scale argument)</p> <p>‘new.xdim’ default: NA; scale X axis to specified number of neurons</p> <p>‘new.ydim’ default: NA; scale Y axis to specified number of neurons</p> <p>‘show.counter.border’ default: 0.98 percentile as limit for the display of labels in the pie charts</p>

**Value**

The function does not returns a value.

**Author(s)**

Benjamin Schulz, Andreas Dominik

**References**

Schulz, B., Mauthe, T., Dominik, A., 2010. Visualisation of Kohonen Maps using R. THM University of Applied Sciences, Giessen, Germany Website, [Online]. Available at: <http://www.life-science-it.org/pages/research/projectSomplot.html> [Accessed 19 May 2011]. and <http://www.bioconductor.org/packages/2.5/bioc/html/hexbin.html>

**Examples**

```
som.plot(system.file("test.data/iris4som.out", package="somplot"),
         system.file("test.data/iris4som.dat", package="somplot"))
```

# Index

\*Topic **Graphics**

hexbinpie, [2](#)

makehexbinplot, [4](#)

som.plot, [5](#)

\*Topic **High-Level Plots**

hexbinpie, [2](#)

makehexbinplot, [4](#)

som.plot, [5](#)

\*Topic **package**

somplot-package, [2](#)

hexbinpie, [2](#)

makehexbinplot, [4](#)

som.plot, [5](#)

somplot (somplot-package), [2](#)

somplot-package, [2](#)