

# Package ‘NeatMap’

September 21, 2009

**Type** Package

**Title** Non-clustered heatmap alternatives

**Version** 0.3.1

**Date** 2009-09-18

**Author** Satwik Rajaram and Yoshi Oono

**Depends** ggplot2, rgl

**Maintainer** Satwik Rajaram <srjaram@illinois.edu>

**Description** NeatMap is a package to create heatmap like plots in 2 and 3 dimensions, without the need for cluster analysis. Like the heatmap, the plots created by NeatMap display both a dimensionally reduced representation of the data as well as the data itself. They are intended to be used in conjunction with dimensional reduction techniques such as PCA.

**License** GPL-3

**LazyLoad** yes

**Repository** CRAN

**Date/Publication** 2009-09-21 06:51:01

## R topics documented:

NeatMap-package . . . . .	2
circularmap . . . . .	4
draw.dendrogram . . . . .	6
draw.dendrogram3d . . . . .	7
heatmap1 . . . . .	9
lineplot . . . . .	11
nMDS . . . . .	12
profileplot3d . . . . .	13
stereo.profileplot3d . . . . .	16

<b>Index</b>	<b>19</b>
--------------	-----------

---

 NeatMap-package      *Non-clustered Heatmaps*


---

## Description

The **NeatMap** package is a set of functions to create heatmap like plots in two and three dimensions, without the need for cluster analysis. Like the heatmap, the plots created by **NeatMap** display both a dimensionally reduced representation of the data as well as the data itself. They are intended to be used in conjunction with dimensional reduction techniques such as PCA (as opposed to cluster analysis for the standard clustered heatmap).

## Details

Package:	NeatMap
Type:	Package
Version:	0.3.1
Date:	2009-09-18
License:	GPL-3
LazyLoad:	yes

The traditional clustered heatmap makes use of cluster analysis to re-order rows and columns such that similar elements are placed together. However, cluster analysis is a poor choice for ordering method since it does not provide a unique ordering. The cluster analysis results are meant to be read along the tree, not in terms of the leaf order. The leaf order may be changed, while preserving the tree structure, by swinging the leaves at the bifurcations. In fact there are methods that exploit this degree of freedom to improve the appearance of the heatmap. However, these are not standardized and could potentially place unrelated elements together. Apart from this, for the results of cluster analysis to be reliable the separation of groups need to be very pronounced. When this is not the case other dimensional reduction methods are likely to provide a better representation of the data. The **NeatMap** package is intended to be used in conjunction with such methods to display both the dimensional reduction result as well as the data underlying it. It includes plots in 2 and 3 dimensions. The two dimensional plots are built using the **ggplot2** package while the three dimensional plots use **rgl**.

The most basic functions are `heatmap1` and its convenience wrapper function `make.heatmap1`. This is virtually identical to the traditional heatmap, except the ordering of rows and columns do not use cluster analysis. The user may either supply an ordering of the rows using the method of their choice, or PCA/nMDS may be used. In the latter case, if normalized data or distance measure are used it is common to get an annular embedding in two dimensions (e.g. first two PCA components). The angular positions in this embedding are then used to order the rows and columns in the heatmap. However, the two opposite ends in this ordering will be separated by 360 degrees and are therefore very symmetric. To avoid artifacts produced by this, one may use the `make.circularmap` function which twists the heatmap into an annular format to reflect this periodicity (see examples below).

The `lineplot` takes the 2 dimensional embedding result, places it in a grid. Then for each grid cell, the profiles of all the points in that cell are displayed together as line plots. Lineplots are easier

to comprehend than intensity patterns, so this format provides a good representation of the data. However, rows and columns are not treated on an equal footing, and comparison of genes is more difficult than the heatmap.

A circle like embedding was required to produce the ordering used in `make.heatmap1` and `make.circularmap`. In the more general case, where the embedding is not circular one may use the 3d plot `profileplot3d` (or its convenience wrapper function `make.profileplot3d`). The 2d embedding of rows is placed in the xy plane. For each point (i.e. row) in the xy plane, its profile, heatmap style, is shown parallel to the z axis. It is possible to rotate and zoom this plot to focus on interesting parts. Stereo versions of these plots may be created using `stereo.profileplot3d` (or its convenience wrapper function `stereo.profileplot3d`). The stereo plots too are dynamically rotatable, and give the impression of observing truly three dimensional structure. Stereo plots could be useful in showing a 3D structure in publications (since rotations will not be possible).

There are also the dendrogram creating functions `draw.dendrogram` and its 3d version `draw.dendrogram3d` which are used internally by the functions described above. However, they may be called directly to compare the cluster analysis result to that of the dimensional reduction method.

For convenience purposes an implementation of non-Metric Multidimensional Scaling is also provided through the `nMDS` function.

### Author(s)

Satwik Rajaram and Yoshi Oono Maintainer: Satwik Rajaram <sraram@illinois.edu>

### See Also

[heatmap1](#), [circularmap](#), [lineplot](#), [profileplot3d](#), [stereo.profileplot3d](#), [draw.dendrogram](#), [draw.dendrogram3d](#).

### Examples

```
#heatmap1 using pca (the scale_x_continuous ensures that labels can be seen)
make.heatmap1(mtcars, row.method="PCA", column.method="average.linkage",
  row.labels=rownames(mtcars), column.labels=colnames(mtcars)) +
  scale_x_continuous(lim=c(-1,15))
#circularmap using nMDS and pearson correlation
make.circularmap(as.matrix(mtcars), metric="euclidean", cluster.method="complete.linkage",
  normalize.profiles=FALSE, label.names=rownames(mtcars), label.size=3)

#lineplot using pca
mtcars.PCA<-prcomp(mtcars);
lineplot(mtcars.PCA$x,mtcars);

#profileplot3d and stereo.profileplot3d using PCA
make.profileplot3d(mtcars, row.method="PCA", column.method="average.linkage")
make.stereo.profileplot3d(mtcars, row.method="PCA", column.method="average.linkage")

#dendrogram3d to compare nMDS result to clustering
mtcars.nMDS<-nMDS(mtcars, metric="euclidean")
mtcars.cluster<-hclust(dist(as.matrix(mtcars)), method="complete")
draw.dendrogram3d(mtcars.cluster,mtcars.nMDS$x, labels=rownames(mtcars),
```

```
label.size=0.5)
```

---

```
circularmap
```

```
Heatmap arranged as an annular region
```

---

## Description

These functions display a matrix as an annular heatmap reflecting the periodicity of the row ordering

## Usage

```
circularmap(pos, profiles, column.order=NULL, cluster.result = NULL,
cluster.heights = NULL, Rin = 10, Rout = 30, thickness = 3, label.names = NULL,
Rlabel = 32, label.size = 1.5, normalize.profiles = T)
```

```
make.circularmap(profiles, method = "nMDS", column.method="none",
cluster.method = "average.linkage", metric = "pearson", column.metric="pearson",
Rin = 10, Rout = 30, thickness = 3, label.names = NULL, Rlabel = 32,
label.size = 1.5, normalize.profiles = T, row.random.seed=NULL, column.random.seed=NULL)
```

## Arguments

<code>pos</code>	The positions for the rows as produced by some dimensional reduction technique. Can either be list of angles, or of 2d positions.
<code>profiles</code>	A matrix containing the values to be displayed as a heatmap.
<code>column.order</code>	A vector containing the order in which columns should be shown
<code>cluster.result</code>	The hierarchical clustering result of type <code>hclust</code> for the rows, superposed on the heatmap for validation. If <code>NULL</code> no clustering result is shown.
<code>cluster.heights</code>	An optional vector of heights to over-ride the heights specified in <code>cluster.result</code> .
<code>Rin</code>	Inner radius of annulus.
<code>Rout</code>	Outer radius of annulus.
<code>thickness</code>	Thickness of the individual row expression level bands.
<code>label.names</code>	list of row labels.
<code>Rlabel</code>	Radius at which row labels are placed.
<code>label.size</code>	Font size for labels.
<code>normalize.profiles</code>	logical: if true the profiles are normalized (mean zero, unit variance) before display.
<code>method</code>	The dimensional reduction method used by <code>make.circularmap</code> to produce angular positions for the rows. One of "nMDS" or "PCA".

<code>column.method</code>	The dimensional reduction method used by <code>make.circularmap</code> to order columns. One of "none", "nMDS", "PCA", "average.linkage" or "complete.linkage".
<code>cluster.method</code>	clustering method used for superposed row cluster. Either "complete.linkage" or "average.linkage".
<code>metric</code>	the distance function used for row embedding. Can be either "pearson" or "euclidean".
<code>column.metric</code>	the distance function used for column embedding. Can be either "pearson" or "euclidean".
<code>row.random.seed</code>	Random seed to be used if nMDS is used to construct row embedding
<code>column.random.seed</code>	Random seed to be used in nMDS is used to generate column ordering

## Details

These are function used to construct heatmap like plots arranged in an annular ring. The assumption is that the 2d embedding result for the rows of the given matrix is circular in shape (the user should confirm this is indeed the case). The angular positions of each point (i.e., row) in this embedding is the angular position of its profile in the circular map. `circularmap` is the base function that takes the expression matrix and the result of dimensional reduction (expressed as a 2d positions or a list of angles) and produces the annular heatmap. `make.circularmap` is a convenience wrapper function that takes the given matrix, performs dimensional reduction using either "nMDS" or "PCA", produces cluster analysis on the row, and passes these results to `circularmap` for plotting.

Apart from the circular (vs linear) ordering, this function differs from `heatmap1` in that the profiles are no longer equally spaced, and may reflect the non-uniformity of the angular positions. On the other hand, `heatmap1` treats rows and columns on an equal footing while these functions focus primarily on the relations between rows.

The cluster analysis makes use of the same distance measure (specified by `metric`) as the dimensional reduction method.

## Value

A `ggplot2` plot of class `ggplot`.

## Author(s)

Satwik Rajaram and Yoshi Oono

## See Also

[image](#), [heatmap](#), [heatmap1](#).

**Examples**

```

make.circularmap(as.matrix(mtcars),metric="euclidean",cluster.method="complete.linkage",
normalize.profiles=FALSE)

#is equivalent to
mtcars.nMDS<-nMDS(as.matrix(mtcars),metric="euclidean")
mtcars.cluster<-hclust(dist(mtcars),method="complete")
circularmap(mtcars.nMDS$x,as.matrix(mtcars),normalize.profiles=FALSE,
cluster.result=mtcars.cluster)

#To change coloring scheme etc
make.circularmap(as.matrix(mtcars),metric="euclidean",cluster.method="complete.linkage",
normalize.profiles=FALSE)+scale_fill_gradient2(low="yellow",high="blue",
mid="black",midpoint=200);

```

---

draw.dendrogram	<i>Draws a dendrogram in 2d</i>
-----------------	---------------------------------

---

**Description**

Draw a dendrogram in 2d given clustering results. Leaf order may be specified, allowing comparison to clustering result.

**Usage**

```
draw.dendrogram(cluster, leaf.order=NULL, scale = 10, dendro.dir = "left",
order.dir = "up", origin = as.vector(c(0.5, 0)), heights = NULL)
```

**Arguments**

cluster	Hierarchical clustering result of type <code>hclust</code> to be plotted.
leaf.order	A vector containing the order of leaf tips, if <code>NULL</code> , the order specified in <code>cluster</code> is used
scale	Maximum dendrogram height. Width is equal to number of leaves
dendro.dir	Direction in which the leaves point. One of "up", "down", "left" or "right"
order.dir	Direction in which leaves are ordered. Should be perpendicular to <code>dendro.dir</code> . One of "up", "down", "left" or "right"
origin	Position of first leaf
heights	A vector of heights, which may be used to over-ride the height information included in <code>cluster</code> (which are used by default if this is <code>NULL</code> )

## Details

Similar to the `plot` command for cluster results of type `hclust`. Primarily intended for internal use, but for ease of formatting, may be of use in producing standard dendrogram plots instead of the standard `hclust` commands. Also the leaf order, and branch heights used here may be different from that specified in `cluster`. The leaf tips are placed at the integer valued positions (with respect to the origin) specified in the leaf order. This may be used to place labels, colored points etc at the leaf tips.

## Value

a `ggplot` layer

## Note

The leaf heights are scaled so that the maximum height corresponds to `scale`.

## Author(s)

Satwik Rajaram and Yoshi Oono

## See Also

[hclust](#).

## Examples

```
#compare complete and average linkage
mtcars.cluster<-hclust(dist(mtcars),method="complete")
mtcars.cluster.avg<-hclust(dist(mtcars),method="average")
ggplot.default()+draw.dendrogram(mtcars.cluster,leaf.order=mtcars.cluster.avg$order)
```

---

`draw.dendrogram3d` *Draws a dendrogram in 3d*

---

## Description

Draw a dendrogram in 3d given clustering results and leaf positions. Labels at leaf tips may be specified

## Usage

```
draw.dendrogram3d(cluster, positions, direction = c(0, 0, -1), scale = NULL,
heights = NULL, labels = NULL, label.colors = NULL, label.size = 3)
```

**Arguments**

<code>cluster</code>	Hierarchical clustering result of type <code>hclust</code> to be plotted.
<code>positions</code>	Positions (in 3d) of leaf tips. If 2d positions are given, these are placed in the <code>z=0</code> plane.
<code>direction</code>	A vector specifying the direction in which the leaves point
<code>scale</code>	Maximum height of dendrogram
<code>heights</code>	A vector of heights, which may be used to over-ride the height information included in <code>cluster</code> (which are used by default if this is <code>NULL</code> )
<code>labels</code>	Vector of text labels to be placed at leaf tips.
<code>label.colors</code>	Vector of colors used used for labels.
<code>label.size</code>	Text size for labels.

**Details**

May be used to validate 2D embedding results with those of cluster analysis. When cluster results and the corresponding 2D embedding are specified, this function places the positions in a rotatable environment with the cluster analysis result superposed on it. This allows the user to understand the relationship between the clustering and embedded results. Labels (or desired colors) may be placed at the leaf tips.

**Value**

This function is called for the side effect it produces. It returns the id number of the last object drawn.

**Author(s)**

Satwik Rajaram and Yoshi Oono

**See Also**

[draw.dendrogram](#)

**Examples**

```
mtcars.nMDS<-nMDS(mtcars,metric="euclidean")
mtcars.cluster<-hclust(dist(mtcars),method="complete")
draw.dendrogram3d(mtcars.cluster,mtcars.nMDS$x,labels=rownames(mtcars),label.size=0.5)
```

---

heatmap1	<i>Make a non-clustered heatmap</i>
----------	-------------------------------------

---

## Description

Makes a heatmap without need for cluster analysis

## Usage

```
heatmap1(profiles, row.order = NULL, column.order = NULL, row.cluster = NULL,
column.cluster = NULL, column.labels = NULL, row.labels = NULL,
column.label.size = 3, row.label.size = 3, row.normalize=F)
```

```
make.heatmap1(profiles, row.method = "nMDS", column.method = "none",
row.metric = "pearson", column.metric = "pearson", row.cluster.method = "average",
column.cluster.method = "average", column.labels = NULL, row.labels = NULL,
row.label.size = 3, column.label.size = 3, row.normalize=F, row.random.seed=NULL, col
```

## Arguments

profiles	matrix: containing the data to be plotted.
row.order	vector containing order of rows such as produced by order
column.order	vector containing order of columns such as produced by order
row.cluster	hierarchical clustering result for rows of type hclust
column.cluster	hierarchical clustering result for columns of type hclust
column.labels	vector of labels for columns
row.labels	vector of labels for rows
column.label.size	size for column label text
row.label.size	size for row label text
row.normalize	logical: If true the rows are normalized to zero mean and unit variance
row.method	dimension reduction method used by make.heatmap1 to order rows. One of "none", "nMDS", "PCA", "complete.linkage" and "average.linkage".
column.method	dimension reduction method used by make.heatmap1 to order columns. One of "none", "nMDS", "PCA", "complete.linkage" and "average.linkage".
row.metric	Distance metric used by row.method either "pearson" or "euclidean"
column.metric	Distance metric used by column.method either "pearson" or "euclidean"

```

row.cluster.method
    Clustering algorithm used for clustering rows. Either "average.linkage" or "complete.linkage". If NULL, no row cluster results are shown
column.cluster.method
    Clustering algorithm used for clustering columns. Either "average.linkage" or "complete.linkage". If NULL, no column cluster results are shown
row.random.seed
    Random seed to be used if nMDS is used to construct row ordering
column.random.seed
    Random seed to be used in nMDS is used to generate column ordering

```

### Details

The traditional heatmap uses clustering to order rows and columns. These functions allow us to use alternate schemes for this ordering. They use the same format as the traditional heatmap, and are therefore similar to `image` and `heatmap`. `heatmap1` assumes the user has already ordered the rows and columns according to the scheme of their choice. `make.heatmap1` is a convenience wrapper which performs ordering using "nMDS", "PCA", or hierarchical clustering. This ordering is passed on to `heatmap1` for plotting. If "nMDS" or "PCA" are chosen as the ordering method, then it is assumed that their 2D embedding is annular in shape. This is often observed if PCA or nMDS (with euclidean distance) is applied to normalized data or Pearson correlation is used with nMDS. Angles measured at the centre of mass are then used for ordering. The user is therefore advised to confirm that such an annular structure is indeed present. Note that the two opposite ends of the ordering are typically separated by 360 degrees and are therefore very similar. To avoid artifacts produced by this, the user could consider using `circularmap` instead. The metric specified by `row.metric` and `column.metric` are also used by the clustering algorithms.

### Value

A `ggplot2` plot of class `ggplot`.

### Author(s)

Satwik Rajaram and Yoshi Oono

### See Also

[image](#), [heatmap](#), [circularmap](#).

### Examples

```

make.heatmap1(mtcars, row.method="PCA", column.method="average.linkage")

#is equivalent to
mtcars.PCA<-prcomp(mtcars)
mtcars.PCA.order<-order(apply(mtcars.PCA$x[,1:2], 1, function(x){atan2(x[1], x[2])}))
mtcars.column.cluster<-hclust(as.dist(1-cor(mtcars)), method="average")
mtcars.row.cluster<-hclust(as.dist(1-cor(t(mtcars))), method="average")
heatmap1(mtcars, row.order=mtcars.PCA.order, column.order=mtcars.column.cluster$order,
row.cluster=mtcars.row.cluster, column.cluster=mtcars.column.cluster)

```

```
#Changing Color Scheme
make.heatmap1(mtcars, row.method="PCA", column.method="average.linkage")+
scale_fill_gradient2(low="yellow", high="blue", mid="black", midpoint=200)

#Adding labels (the scale function ensures that labels are not clipped)
make.heatmap1(mtcars, row.method="PCA", column.method="average.linkage",
row.labels=rownames(mtcars), column.labels=colnames(mtcars))+
scale_x_continuous(lim=c(-1, 15))
```

---

lineplot

---

*Line plots of profiles in a grid of embedded results*


---

### Description

A 2d embedding of rows of given matrix is gridded and line plots of the profiles of the points within each cell are displayed

### Usage

```
lineplot(pos, profiles, n.div.x = 10, n.div.y = 10, normalize = F,
ylim=NULL, clipped=F)
```

### Arguments

<code>pos</code>	2d positions of rows
<code>profiles</code>	Matrix of data to be plotted
<code>n.div.x</code>	Number of grid boxes in x direction
<code>n.div.y</code>	Number of grid boxes in y direction
<code>normalize</code>	logical: if true, the profiles are normalized before plotting
<code>ylim</code>	a vector of length 2 containing the <code>profile</code> values that correspond to the top and bottom of the grid boxes, if <code>NULL</code> the largest and smallest values in <code>profile</code> are used
<code>clipped</code>	logical: If <code>TRUE</code> , then values exceeding <code>ylim</code> are clipped

### Details

`pos` is assumed to be the embedding/2 dimensional representation of the rows of `profiles`. The `pos` result is then placed in a uniform grid with the number of divisions in the x and y directions specified by `n.div.x` and `n.div.y` respectively. Then in each grid cell the profiles for all the points in it are displayed together. Missing data is not plotted. The grid extends 5 percent of the difference between the max and min point beyond these points. In each cell 90 percent of its width is used. By default, the profiles are scaled so that the maximum and minimum values (in `profiles`) would appear at the top and bottom of a cell. Different limits can be chosen using the `ylim` option. If `clipped` is true values going beyond these limits will be clipped, in order to prevent the overlap of profiles in different cells. If `normalize` is true, the profiles shall be normalized to have zero mean and unit variance.

**Value**

Returns a `ggplot2` plot of class `ggplot`.

**Note**

`ylim` is applied to the profiles that will be plotted. So if normalization is turned on, the limits apply to these normalized values which are not accessible to the user. Therefore if this functionality is desired it may be better to normalize the data before invoking `lineplot`.

**Author(s)**

Satwik Rajaram and Yoshi Oono

**Examples**

```
#PCA and line plot. Notice how the profiles are dominated by two high value columns
mtcars.PCA<-prcomp(mtcars);
lineplot(mtcars.PCA$x,mtcars);

#Use ylim and clipping to allow us to focus on the columns with lower values
lineplot(mtcars.PCA$x,mtcars,ylim=c(0,10),clipped=TRUE)
```

---

nMDS

*non-Metric Multi-Dimensional Scaling*


---

**Description**

Given a matrix, and a distance measure, an embedding of the rows into desired Euclidean space is performed using non-Metric Multi-Dimensional Scaling.

**Usage**

```
nMDS(data, embed.dim = 2, n.iters = 300, metric = "pearson", random.seed=NULL)
```

**Arguments**

<code>data</code>	matrix whose rows shall be embedded.
<code>embed.dim</code>	Dimensionality of Euclidean space into which embedding shall be performed.
<code>n.iters</code>	Number of iterations of the nMDS scheme
<code>metric</code>	The distance metric used to compare rows. Currently only "pearson" and "euclidean" are supported.
<code>random.seed</code>	A random seed used by nMDS. Use of this option allows reproducibility of nMDS results

## Details

non-Metric Multi-Dimensional Scaling is performed using the scheme proposed by Taguchi and Oono.

If an element is missing (NA) in a particular row, all distance comparisons to that row shall ignore that particular element.

## Value

An object of class "nMDS" containing:

x                    matrix with the same number of rows and row names as data and having  
embed.data columns

## Author(s)

Satwik Rajaram and Yoshi Oono

## References

Relational patterns of gene expression via non-metric multidimensional scaling analysis: Y.-h. Taguchi and Y. Oono, *Bioinformatics*, 2005 21(6):730-740.

## See Also

[prcomp](#)

## Examples

```
#Two dimensional embedding
mtcars.nMDS<-nMDS(as.matrix(mtcars),embed.dim=2,metric="euclidean")
plot(mtcars.nMDS$x,type='n')
text(mtcars.nMDS$x,labels=rownames(mtcars.nMDS$x))
```

---

profileplot3d

*Make a 3D rotatable plot showing data profiles*

---

## Description

Make a 3d rotatable plot depicting the intensity levels of a matrix, while showing the relations between rows in two dimensions and that of the columns in the third. Cluster analysis results for rows and columns may be superposed.

**Usage**

```
profileplot3d(pos, profiles, normalize.rows = T, column.order = NULL,
row.cluster = NULL, column.cluster = NULL, labels = NULL, col = NULL,
color_scaling_function=NULL, point.size = 3, label.colors = NULL,
label.size = 0.5)
```

```
make.profileplot3d(profiles, row.method = "nMDS", normalize.rows=T,
column.method = "average.linkage", row.metric = "pearson",
column.metric = "pearson", row.cluster.method = "average",
column.cluster.method = "average", point.size = 3, col=NULL,
color_scaling_function=NULL, labels = NULL, label.colors = NULL,
label.size = 0.5, row.random.seed=NULL, column.random.seed=NULL)
```

**Arguments**

<code>pos</code>	matrix: the 2d positions for the rows in profiles as produced by any dimensional reduction scheme.
<code>profiles</code>	matrix: containing the data to be plotted.
<code>normalize.rows</code>	logical: If TRUE, then the rows shall be normalized before plotting.
<code>column.order</code>	The ordering of the columns, as would be the case in a typical heatmap, produced using some dimensional reduction scheme. If it is NULL, then the ordering in profiles is used.
<code>row.cluster</code>	hierarchical clustering result (of type <code>hclust</code> ), of the rows, for superimposing the clustering result on the 3d profile plot. If it is NULL, no cluster result will be plotted.
<code>column.cluster</code>	similar to <code>row.cluster</code> except for the clustering of columns.
<code>labels</code>	labels for the rows. If it is set to NULL, no labels will be plotted.
<code>col</code>	A list of colors such as that generated by <code>rainbow</code> used in depicting low to high intensities as in a heat-map.
<code>color_scaling_function</code>	A function mapping [0:1] onto [0:1] used for scaling the color levels. If Null, linear scaling is performed
<code>point.size</code>	The size of intensity points.
<code>label.colors</code>	A list of colors used for the row labels
<code>label.size</code>	Initial size of row labels. The sizes can be changed dynamically after plotting using the 3rd mouse button
<code>row.method</code>	dimensional reduction method for embedding rows, currently only "PCA" and "nMDS" are supported.
<code>column.method</code>	dimensional reduction method for ordering columns. Can be any one of "nMDS", "PCA", "average.linkage" and "complete.linkage".
<code>row.metric</code>	the distance function used for row embedding. Can be either "pearson" or "euclidean".

```

column.metric           like row.metric except for columns.
row.cluster.method     clustering method used for superposed row cluster. Either "complete.linkage"
                        or "average.linkage".
column.cluster.method  like row.metric except for columns.
row.random.seed        Random seed to be used if nMDS is used to construct row structure
column.random.seed    Random seed to be used in nMDS is used to generate column ordering

```

### Details

These functions display data in a 3d rotatable format. The xy positions are the result of a 2D embedding of the rows, the profiles for which are shown along the z-axis. `profileplot3d` is the primary function to do this, accepting the results of any dimensional reduction scheme, while `make.profileplot3d` is a convenience function performing both the dimensional reduction (using nMDS or PCA) and then calling `profileplot3d`.

The ordering of columns is similar to that for `heatmap1`. Thus if PCA or nMDS are used it is assumed that the embedding of columns is annular, and the order used is that of angular positions. The user should confirm that this is indeed the case.

### Value

These functions are called for the side-effects they produce

### Author(s)

Satwik Rajaram and Yoshi Oono

### See Also

[heatmap1](#).

### Examples

```

make.profileplot3d(mtcars, row.method="PCA", column.method="average.linkage")

#is equivalent to
mtcars.PCA<-prcomp(mtcars)
mtcars.col.cluster<-hclust(dist(t(mtcars)), method="average")
mtcars.row.cluster<-hclust(as.dist(1-cor(t(mtcars))), method="average")
profileplot3d(mtcars.PCA$x, mtcars, column.order=mtcars.col.cluster$order,
row.cluster=mtcars.row.cluster, column.cluster=mtcars.col.cluster)

#use of alternate colors and color scaling
make.profileplot3d(mtcars, row.method="PCA", column.method="average.linkage",
col=c("yellow", "black", "blue"),
color_scaling_function=function(x){0.5+tanh(10*(x-0.5))/2})

```

---

```
stereo.profileplot3d
```

*Make a Stereo rotatable plot showing data profiles*

---

## Description

Make a stereo rotatable plot depicting the intensity levels of a matrix, while showing the relations between rows in two dimensions and that of the columns in the third. Cluster analysis results for rows and columns may be superposed. Stereo version of `profileplot3d`

## Usage

```
stereo.profileplot3d(pos, profiles, normalize.rows = T, column.order = NULL,
row.cluster = NULL, column.cluster = NULL, labels = NULL, col = NULL,
color_scaling_function=NULL, point.size = 3, label.colors = NULL,
label.size = 0.5, stereo.angle=5)
```

```
make.stereo.profileplot3d(profiles, row.method = "nMDS", normalize.rows=T,
column.method = "average.linkage", row.metric = "pearson",
column.metric = "pearson", row.cluster.method = "average",
column.cluster.method = "average", point.size = 3, col=NULL,
color_scaling_function=NULL, labels = NULL, label.colors = NULL,
label.size = 0.5, row.random.seed=NULL, column.random.seed=NULL, stereo.angle=5)
```

## Arguments

<code>pos</code>	matrix: the 2d positions for the rows in profiles as produced by any dimensional reduction scheme.
<code>profiles</code>	matrix: containing the data to be plotted.
<code>normalize.rows</code>	logical: If TRUE, then the rows shall be normalized before plotting.
<code>column.order</code>	The ordering of the columns, as would be the case in a typical heatmap, produced using some dimensional reduction scheme. If it is NULL, then the ordering in profiles is used.
<code>row.cluster</code>	hierarchical clustering result (of type <code>hclust</code> ), of the rows, for superimposing the clustering result on the 3d profile plot. If it is NULL, no cluster result will be plotted.
<code>column.cluster</code>	similar to <code>row.cluster</code> except for the clustering of columns.
<code>labels</code>	labels for the rows. If it is set to NULL, no labels will be plotted.
<code>col</code>	A list of colors such as that generated by <code>rainbow</code> used in depicting low to high intensities as in a heat-map.
<code>color_scaling_function</code>	A function mapping [0:1] onto [0:1] used for scaling the color levels. If Null, linear scaling is performed

<code>point.size</code>	The size of intensity points.
<code>label.colors</code>	A list of colors used for the row labels
<code>label.size</code>	size of row labels.
<code>stereo.angle</code>	Difference in perspective angle (in degrees) between the two stereo figures
<code>row.method</code>	dimensional reduction method for embedding rows, currently only "PCA" and "nMDS" are supported.
<code>column.method</code>	dimensional reduction method for ordering columns. Can be any one of "nMDS", "PCA", "average.linkage" and "complete.linkage".
<code>row.metric</code>	the distance function used for row embedding. Can be either "pearson" or "euclidean".
<code>column.metric</code>	like <code>row.metric</code> except for columns.
<code>row.cluster.method</code>	clustering method used for superposed row cluster. Either "complete.linkage" or "average.linkage".
<code>column.cluster.method</code>	like <code>row.metric</code> except for columns.
<code>row.random.seed</code>	Random seed to be used if nMDS is used to construct row structure
<code>column.random.seed</code>	Random seed to be used in nMDS is used to generate column ordering

### Details

These functions display data in a 3d rotatable stereo format. They are just stereo versions of the functions `profileplot3d` and `make.profileplot3d`. The xy positions are the result of a 2D embedding of the rows, the profiles for which are shown along the z-axis. `stereo.profileplot3d` is the primary function to do this, accepting the results of any dimensional reduction scheme, while `make.stereo.profileplot3d` is a convenience function performing both the dimensional reduction (using nMDS or PCA) and then calling `profileplot3d`.

The ordering of columns is similar to that for `heatmap1`. Thus if PCA or nMDS are used it is assumed that the embedding of columns is annular, and the order used is that of angular positions. The user should confirm that this is indeed the case.

### Value

These functions are called for the side-effects they produce

### Author(s)

Satwik Rajaram and Yoshi Oono

### See Also

[heatmap1](#).

**Examples**

```
make.stereo.profileplot3d(mtcars, row.method="PCA", column.method="average.linkage")

#is equivalent to
mtcars.PCA<-prcomp(mtcars)
mtcars.col.cluster<-hclust(dist(t(mtcars)), method="average")
mtcars.row.cluster<-hclust(as.dist(1-cor(t(mtcars))), method="average")
stereo.profileplot3d(mtcars.PCA$x, mtcars, column.order=mtcars.col.cluster$order,
row.cluster=mtcars.row.cluster, column.cluster=mtcars.col.cluster)

#use of alternate colors and color scaling
make.stereo.profileplot3d(mtcars, row.method="PCA", column.method="average.linkage",
col=c("yellow", "black", "blue"),
color_scaling_function=function(x){0.5+tanh(10*(x-0.5))/2})
```

# Index

- \*Topic **aplot**
  - draw.dendrogram, 6
- \*Topic **hplot**
  - circularmap, 4
  - draw.dendrogram3d, 7
  - heatmap1, 9
  - lineplot, 11
  - profileplot3d, 13
  - stereo.profileplot3d, 16
- \*Topic **multivariate**
  - nMDS, 12
- \*Topic **nonparametric**
  - nMDS, 12
- \*Topic **package**
  - NeatMap-package, 2

circularmap, 3, 4, 10

draw.dendrogram, 3, 6, 8

draw.dendrogram3d, 3, 7

hclust, 7

heatmap, 5, 10

heatmap1, 3, 5, 9, 15, 17

image, 5, 10

lineplot, 3, 11

make.circularmap(*circularmap*), 4

make.heatmap1(*heatmap1*), 9

make.profileplot3d, 17

make.profileplot3d  
(*profileplot3d*), 13

make.stereo.profileplot3d  
(*stereo.profileplot3d*), 16

NeatMap(*NeatMap-package*), 2

NeatMap-package, 2

nMDS, 12

prcomp, 13

profileplot3d, 3, 13, 17

stereo.profileplot3d, 3, 16