

Package ‘idendr0’

October 16, 2015

Type Package

Title Interactive Dendrograms

Version 1.5.2

Date 2015-10-16

Author Tomas Sieger

Depends

Imports tcltk, tkrplot, grDevices, graphics, stats

Suggests rggobi, DendSer, cluster, RColorBrewer, datasets

URL <http://github.com/tsieger/idendr0>

Maintainer Tomas Sieger <tomas.sieger@seznam.cz>

BugReports <https://github.com/tsieger/idendr0/issues>

Description Interactive dendrogram that enables the user to select and color clusters, to zoom and pan the dendrogram, and to visualize the clustered data not only in a built-in heat map, but also in 'GGobi' interactive plots and user-supplied plots.
This is a backport of Qt-based 'idendro'
(<https://github.com/tsieger/idendro>) to base R graphics and Tcl/Tk GUI.

License GPL-2

LazyLoad yes

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2015-10-16 20:30:35

R topics documented:

idendr0-package	2
hca5000	2
idendro	3

Index	8
--------------	----------

idendr0-package *Interactive Dendrograms*

Description

Interactive dendrogram that enables the user to select and color clusters, to zoom and pan the dendrogram, and to visualize the clustered data not only in a built-in heat map, but also in 'GGobi' interactive plots and user-supplied plots. This is a backport of Qt-based 'idendro' (<https://github.com/tsieger/idendro>) to base R graphics and Tcl/Tk GUI.

Details

Package: idendr0
 Type: Package
 Title: Interactive Dendrograms
 Version: 1.5.2
 Date: 2015-10-16
 Author: Tomas Sieger
 Depends:
 Imports: tcltk, tkrplot, grDevices, graphics, stats
 Suggests: rggobi, DendSer, cluster, RColorBrewer, datasets
 URL: <http://github.com/tsieger/idendr0>
 Maintainer: Tomas Sieger <tomas.sieger@seznam.cz>
 BugReports: <https://github.com/tsieger/idendr0/issues>
 License: GPL-2
 LazyLoad: yes
 LazyData: true

Author(s)

Tomas Sieger

hca5000 *idendro demo data*

Description

Hierarchical cluster analysis demonstration data consisting of raw data (5000 observations having 3 features measured) and clustered data (as computed by 'hclust').

Usage

```
data(hca5000)
```

Format

A list of 'x' and 'hx' elements. 'x' is a matrix of 5000 rows (observations) by 3 columns (dimensions). 'hx' is an object of class 'hclust' containing the result of hierarchical cluster analysis performed on 'x'.

Examples

```
if (interactive()) {
  library(idendro)
  data(hca5000)
  idendro(hca5000$hx, hca5000$x, observationAnnotationEnabled = FALSE)
}
```

 idendro

Interactive Dendrogram

Description

'idendro' is a plot enabling users to visualize a dendrogram and inspect it interactively: to select and color clusters anywhere in the dendrogram, to zoom and pan the dendrogram, and to visualize the clustered data not only in a built-in heat map, but also in any interactive plot implemented in GGobi (as available using the 'rggobi' package). The integration with GGobi (enabled using the 'ggobi' argument), but also with the user's code is implemented in terms of two callbacks (see the 'colorizeCallback' and 'fetchSelectedCallback' arguments). 'idendro' can be used to inspect quite large dendrograms (tens of thousands of observations, at least).

The 'idendro' package is a lightweight backport of the 'idendro' package. While the 'idendro' package depends on libraries not easily available on some platforms (e.g. Windows), the 'idendro' package is based on platform-independent Tcl/Tk graphic widget toolkit, and thus made widely available. However, the 'idendro' package should be preferred, if available, for its better interactivity and performance.

Usage

```
idendro(h, x = NULL, qx = NULL, clusters = NULL, hscale = 1.5,
  vscale = 1.5, silent = FALSE, zoomFactor = 1/240,
  observationAnnotationEnabled = TRUE,
  clusterColors = c("red", "green", "blue", "yellow", "magenta",
    "cyan", "darkred", "darkgreen", "purple", "darkcyan"),
  unselectedClusterColor = "black", maxClusterCount = max(length(clusterColors),
    ifelse(!is.null(clusters), max(clusters), 0)), heatmapEnabled = TRUE,
  heatmapSmoothing = c("none", "cluster", "zoom"),
  heatmapColors = colorRampPalette(c("#00007F", "blue", "#007FFF", "cyan",
    "#7FFF7F", "yellow", "#FF7F00", "red", "#7F0000"))(10),
```

```
doScaleHeatmap = TRUE, doScaleHeatmapByRows = FALSE,
heatmapRelSize = 0.2, colorizeCallback = NULL, fetchSelectedCallback = NULL,
brushedmapEnabled = !is.null(fetchSelectedCallback),
brushedmapRelSize = ifelse(!is.null(x), heatmapRelSize/ncol(x), 0.05),
geometry = NULL, ggobi = FALSE, ggobiGlyphType = 1, ggobiGlyphSize = 1,
ggobiFetchingStyle = "selected", ggobiColorScheme = "Paired 12", dbg = 0, ...)
```

Arguments

<code>h</code>	object of class <code>'stats::hclust'</code> (or other class convertible to class <code>'hclust'</code> by the <code>'as.hclust'</code> function) describing a hierarchical clustering. If <code>_inversions_</code> in heights (see <code>'hclust'</code>) is detected, the heights get fixed in a simple naive way by preserving non-negative relative differences in the heights, but changing negative differences to zero. Using clustering with monotone distance measure should be considered in that case.
<code>x</code>	data frame holding observations that were clustered giving rise to <code>'h'</code> . The heat map will depict this data. (The heat map can be scaled - see the <code>'doScaleHeatmap'</code> and <code>'doScaleHeatmapByRows'</code> arguments.) Non-numeric types will get converted to numeric using <code>'as.numeric'</code> . This parameter is optional.
<code>qx</code>	(unused, appears for compatibility with <code>idendro::idendro</code>).
<code>clusters</code>	the assignment of observations to clusters to start with, typically the value of a previous call to <code>'idendro'</code> . A numeric vector of length of the number of observations is expected, in which 0s denote unselected observations, and values of $i > 0$ mark members of the cluster <code>'i'</code> .
<code>hscale</code>	horizontal scaling factor of the dendrogram figure. As the dendrogram is implemented as a Tcl/Tk image, and <code>rtcltk</code> does not support image resizing (e.g. on window maximization), the dendrogram keeps its original size regardless of the size of its enclosing window. Thus specifying the <code>hscale</code> of more than 100% is preferred to make the dendrogram large enough.
<code>vscale</code>	vertical scaling factor of the dendrogram figure. See <code>'hscale'</code> .
<code>silent</code>	if TRUE, no informative messages will be shown
<code>zoomFactor</code>	the amount of zoom in/out as controlled by the mouse wheel
<code>observationAnnotationEnabled</code>	shall the names of individual observations (rownames of <code>'x'</code>) be shown next to the dendrogram/heat map?
<code>clusterColors</code>	colors of individual clusters
<code>unselectedClusterColor</code>	the color of unselected dendrogram branches
<code>maxClusterCount</code>	the maximum number of clusters user can select. If greater than the number of <code>'clusterColors'</code> , cluster colors will get recycled. This parameter affects the size of the GUI and the number of clusters that can be selected automatically by "cutting" the dendrogram.
<code>heatmapEnabled</code>	shall the heat map be drawn?

heatmapSmoothing	heat map smoothing mode, one of 'none' - the heat map gets never smoothed, it displays the features of all the individual observations 'cluster' - the heat map depicts the average features for the currently selected clusters, 'zoom' - the heat map displays the average feature for each elementary (i.e. the finest) cluster seen in the dendrogram currently.
heatmapColors	heat map color palette represented by a list of colors, e.g. a sequential palette generated by 'brewer.pal', or 'colorRampPalette(.)', 'gray.colors(.)', or 'hsv(.)'.
doScaleHeatmap	scale each heat map column to the <0,1> range? (The default is TRUE.)
doScaleHeatmapByRows	scale heat map rows, not columns (The default is FALSE.)
heatmapRelSize	relative size of the heat map - the ratio of the heat map width to the width of the dendrogram, the heat map, and the brushed map. The default is 20%.
colorizeCallback	callback function called when cluster selection changes; the argument is a vector assigning color indices (0=no color, >0 colors) to individual observations.
fetchSelectedCallback	callback function used to fetch observation selection made externally. The callback must return a boolean vector of length of the number of observations in 'x'. i-th element in the vector specifies whether given observation is selected.
brushedmapEnabled	shall brushed map be drawn? If TRUE, a column vector is drawn next to dendrogram (and heatmap, if there is one) depicting observation that were fetched by fetchSelectedCallback. The color of the observations is the color of the cluster used to fetch observations into.
brushedmapRelSize	relative size of the brushed map - the ratio of the brushed map width to the width of the dendrogram, the heat map, and the brushed map. The default is the size of a single column in the heat map, or 5% if there is no heatmap.
geometry	window geometry ("width x height + xoffset + yoffset"). Almost useless as the dendrogram does not resize, see the 'hscale' and 'vscale' arguments instead.
ggobi	plot feature space projections of 'x' in ggobi and bidirectionally integrate with the plot? (defaults to FALSE as some users may not have ggobi available)
ggobiGlyphType	ggobi glyph type used to draw observations in ggobi (defaults to a single pixel; see rggobi::glyph_type)
ggobiGlyphSize	size of ggobi glyphs (see rggobi::glyph_size)
ggobiFetchingStyle	how should we recognize ggobi-selected observations to be fetched to idendro? Use 'selected' to fetch observations selected by ggobi brush, or glyph type number 2-6 to fetch observations selected by ggobi persistent brushing with a specific glyph type.
ggobiColorScheme	GGobi color scheme used to color observations in ggobi plots according to the clusters selected in the dendrogram
dbg	debug level (0=none, 1=brief, 2=verbose)
...	additional graphical parameters to be passed to the dendrogram plot.

Details

'idendro' displays an interactive dendrogram enriched, optionally, with a heat map and/or a brushed map.

The dendrogram represents the result of a hierarchical cluster analysis performed on a set of observations (see e.g. 'hclust'). There is an axis drawn below the dendrogram displaying the "height" of the clusters in the dendrogram.

The heat map visualizes the observations living in k-dimensional feature space by mapping their features onto a color scale and displaying them as rows of 'k' colored rectangles. By default, normalization (scaling) of individual features to a common visual scale is enabled. Scaling of observations is also supported (see the 'doScaleHeatmapByRows' argument).

The brushed map can indicate which observations are currently selected in some external plot/tool 'idendro' is integrated with (e.g. a GGobi scatter plot matrix). Technically speaking, the current selection must be determined explicitly by clicking the "fetCh selected" button (or pressing the 'Alt+C' shortcut), which results in calling the 'fetchSelectedCallback' function (see arguments).

The dendrogram can be zoomed and panned. To zoom in a specific region, right click and drag in the dendrogram. Mouse wheel can also be used to zoom in and out. To pan a zoomed dendrogram, middle click and drag the mouse. Zooming and panning history is available (see 'GUI').

User can select clusters manually one by one (by clicking at individual clusters in the dendrogram), or automatically by "cutting" the dendrogram at a specified height. To cut the dendrogram, navigate the mouse close to the dendrogram axis (a dashed line will appear across the dendrogram at a specified height), and left click. Clusters just beneath the cutting height will get selected, replacing the clusters currently selected. Selection history is available (see 'GUI').

Graphic User interface (GUI):

In the left part of the dendrogram window, there is a simple GUI. In the top part of the GUI come cluster-specific controls and info panels arranged in rows. (The number of rows is determined by the 'maxClusterCount' argument.) In each row, there is the current cluster selector (a radio button decorated with a cluster ID and a color code (determined by the 'clusterColors' argument)), and cluster-specific statistics: the total number (and the ratio) of the observations in that specific cluster out of the total number of observations, and the number (and the ratio) of the observations in that cluster out of the observations brushed. The current cluster determines which color and ID will be associated with a cluster selected in the dendrogram, At any time, exactly one cluster is selected as the current cluster.

At the bottom of the GUI window, there are buttons controlling zooming, cluster selection, and heat map smoothing:

"Undo zoom" - retrieves the previous zoom region from history

"Full view" - zooms the dendrogram out maximally

"Undo selection" - retrieves the previous cluster selection from history

"Unselect" - unselects the current cluster in the dendrogram

"Unselect all" - unselects all clusters

The "heat map smoothing" mode can be set to one of:

"none" - the heat map gets never smoothed, it displays the features of all the individual observations

"cluster" - the heat map displays the average features for the currently selected clusters

"zoom" - the heat map displays the average feature for each elementary (i.e. the finest) cluster seen in the dendrogram currently. When the dendrogram is zoomed out maximally, the features of all the elementary clusters (i.e. the individual observations) are displayed. When the user zooms in the dendrogram, such that some clusters get hidden, the features of the observations forming the hidden clusters get averaged.

"Quit"

Value

vector of colors assigned to observations. 0s denote unselected observations, while values of $i > 0$ denote the cluster 'i'.

Author(s)

Tomas Sieger

See Also

idendro::idendro, stats::hclust, stats::plot.hclust

Examples

```
if (interactive()) {
  data(iris, envir = environment())
  hc <- hclust(dist(iris[, 1:4]))
  idendro(hc, iris)
}
# see demos for more examples
```

Index

*Topic **datasets**

hca5000, [2](#)

*Topic **package**

idendr0-package, [2](#)

hca5000, [2](#)

idendr0 (idendr0-package), [2](#)

idendr0-package, [2](#)

idendro, [3](#)