Package 'hierarchicalSets'

May 8, 2026

Type Package

Title Set Data Visualization Using Hierarchies

Version 1.0.4

Author Thomas Lin Pedersen

Maintainer Thomas Lin Pedersen <thomasp85@gmail.com>

Description Pure set data visualization approaches are often limited in scalability due to the combinatorial explosion of distinct set families as the number of sets under investigation increases. hierarchicalSets applies a set centric hierarchical clustering of the sets under investigation and uses this hierarchy as a basis for a range of scalable visual representations. hierarchicalSets is especially well suited for collections of sets that describe comparable comparable entities as it relies on the sets to have a meaningful relational structure.

License MIT + file LICENSE

Encoding UTF-8

LazyData TRUE

Depends R (>= 3.2.0)

Imports ggdendro, ggplot2 (>= 2.2.0), stats, Rcpp, scales, Matrix, MASS, RColorBrewer, gtable, grDevices, methods, utils, viridis, patchwork

RoxygenNote 7.2.3

Collate 'RcppExports.R' 'geom_rect2.R' 'hSet.R' 'hierarchicalSet.R' 'hierarchicalSets-package.R' 'twitter.R' 'visuals.R'

LinkingTo Rcpp

NeedsCompilation yes

Repository CRAN

Date/Publication 2023-04-24 12:50:02 UTC

Contents

`create_hierarchy` *Create and store hierarchcical sets*

Description

HierarchicalSet object can be created using the `hSet()` constructor. The resulting object will contain both the underlying sets as well as the resulting clustering.

Usage

```
create_hierarchy(sets, intersectLimit = 1)

## S3 method for class 'HierarchicalSet'
print(x, ...)

## S3 method for class 'HierarchicalSet'
x[[i]]

## S3 method for class 'HierarchicalSet'
x[i]

## S3 method for class 'HierarchicalSet'
sets(x)

## S3 method for class 'HierarchicalSet'
clusters(x)

## S3 method for class 'HierarchicalSet'
set_names(x)

## S3 method for class 'HierarchicalSet'
element_names(x)

## S3 method for class 'HierarchicalSet'
```

```
n_sets(x)

## S3 method for class 'HierarchicalSet'
length(x)

## S3 method for class 'HierarchicalSet'
n_elements(x)

## S3 method for class 'HierarchicalSet'
n_clusters(x)

## S3 method for class 'HierarchicalSet'
cluster_sizes(x)

## S3 method for class 'HierarchicalSet'
set_sizes(x)

## S3 method for class 'HierarchicalSet'
cluster_members(x)

## S3 method for class 'HierarchicalSet'
set_membership(x)
```

Arguments

`sets`	The sets to analyse. Can either be a matrix/data.frame giving the presence/absence pattern of elements, with elements as rows and sets as columns, or a list of vectors giving the elements of the individual sets.
`intersectLimit`	The proportion of sets an element must be present in to be considered part of the intersect. Standard intersects require it to be present in all sets (`intersectLimit = 1`), which is also the default
`x`	A HierarchicalSet object
`...`	Currently ignored
`i`	The index of the dendrogram

Details

The HierarchicalSet class contains both the clustering and the original sets. The former is stored in a list of dendrogram objects in and the latter as a presence/absence matrix. Both are retrivable using `$clusters` and `$sets` respectively. Furthermore individual dendrograms can be extracted directly using the `[[` operator. If multiple independent clusters exists the object can be subsetted using the `[` operator.

For plotting functionality see the separate plot documentation for `plot.HierarchicalSet()`.

Value

An object of class HierarchicalSet

Functions

- `print(HierarchicalSet)`: Print method for HierarchicalSet objects
- `[[`: Extract dendrogram objects from HierarchicalSet objects
- `[`: Subset HierarchicalSet object by dendrogram (preserving set information and class)
- `sets(HierarchicalSet)`: Extract the sets as a sparse matrix
- `clusters(HierarchicalSet)`: Extract the clusters as a list of dendrograms
- `set_names(HierarchicalSet)`: Get the names of the sets
- `element_names(HierarchicalSet)`: Get the names of the elements
- `n_sets(HierarchicalSet)`: Get the number of sets
- `length(HierarchicalSet)`: Get the number of sets
- `n_elements(HierarchicalSet)`: Get the number of elements
- `n_clusters(HierarchicalSet)`: Get the number of clusters
- `cluster_sizes(HierarchicalSet)`: Get the size of each clusters
- `set_sizes(HierarchicalSet)`: Get the size of each set
- `cluster_members(HierarchicalSet)`: Get the members of each clusters
- `set_membership(HierarchicalSet)`: Get the membership of each set

Examples

```
data('twitter')

# Caclulate the clustering
twitSet <- create_hierarchy(twitter)

# Some statistics on the data
n_sets(twitSet)
n_elements(twitSet)
n_clusters(twitSet)

# Focus on the first two independent cluster
twitSet[1:2]

# Extract a dendrogram representation of the firrst cluster
twitSet[[1]]
```

`HierarchicalSet-getters`

Getters for HierarchicalSet objects

Description

These utility functions makes it easy to extract raw information from a HierarchicalSet object.

Usage

```
sets(x)

clusters(x)

set_names(x)

element_names(x)

n_sets(x)

n_elements(x)

n_clusters(x)

cluster_sizes(x)

set_sizes(x)

cluster_members(x)

set_membership(x)
```

Arguments

x	A HierarchicalSet object

Details

`sets` Returns a ngCMatrix with sets as columns and elements as rows.

`clusters` returns a list of dendrograms with the clustering in the HierarchicalSet object

`set_names` returns a character vector with the names of the sets.

`element_names` returns a character vector with the names of the elements

`n_sets` returns the number of sets

`n_elements` returns the number of elements

`n_clusters` returns the number of independent set families

`cluster_sizes` returns the number of sets in each independent set family

`set_sizes` returns the number of elements in each set

`cluster_members` returns the members of each independent set family

`set_membership` returns the cluster each set is member of

Value

depending on the function. See details

Examples

```
data('twitter')

twitSet <- create_hierarchy(twitter)

# Get the sets as a presence/absence matrix
head(sets(twitSet))

# Get the clustering of the HierarchicalSet object
clusters(twitSet)

# Get the set names
set_names(twitSet)

# Get the element names or NULL if they are unnamed
element_names(twitSet)

# Get the number of sets
n_sets(twitSet)

# Get the number of elements
n_elements(twitSet)

# Get the number of independent clusters
n_clusters(twitSet)

# Get the size of each independent clusters
cluster_sizes(twitSet)

# Get the size of each set
set_sizes(twitSet)

# Get the members of each independent clusters
cluster_members(twitSet)

# Get the membership of each set
set_membership(twitSet)
```

`hierarchicalSets` *Hierarchical analysis and visualization of set data*

Description

This package provides a framework for investigating large scale set data with the use of hierarchical clustering. While hierarchical clustering has been employed on set data numerous times, by converting the presence/absence matrix to a distance matrix and using `stats::hclust()`, this approach completely removes any notion of underlying set structure from the data. hierarchicalSets instead performs a clustering directly using set algebra by continuously merging sets with the largest

intersection (for ties the one with the smallest union is chosen). This structure can then be used in a variety of ways to visualize the relationships between sets. E.g. the intersectionStack plot is a scalable pendant to Venn diagrams (showing the same information but using a different visual mapping).

See Also

`create_hierarchy()` For constructing HierarchicalSet object and `plot.HierarchicalSet()` for visualization apporaches.

`outlier_hierarchy` *Create a new hierarchy based on the outlying elements*

Description

This function detects the outlying elements of a HierarchicalSet object and creates a new clustering of the sets only based on these elements. The returned HierarchicalSet object will only contain the outlying elements, thus reducing the universe size. This operation is somewhat similar to principal component analysis, in that the derived clustering is based on the structure not captured by the first clustering, thus modeling the second most dominant feature of the data.

Usage

```
outlier_hierarchy(set, intersectLimit = 1)
```

Arguments

`set`	A HierarchicalSet object
`intersectLimit`	The proportion of sets an element must be present in to be considered part of the intersect. Standard intersects require it to be present in all sets (intersectLimit = 1), which is also the default

Value

An object of class HierarchicalSet, based on the outliying elements of `set`

See Also

`outlying_elements()` for extracting outlying element information from a HierarchicalSet object

Examples

```
data('twitter')

twitSet <- create_hierarchy(twitter)
twitSetOut <- outlier_hierarchy(twitSet)
twitSetOut
```

`outlying_elements` *Extract the outlying elements from each set pair*

Description

This function detects the outlying elements of each pair of sets in a HierarchicalSet object. An outlying element is defined as an element in the intersection of the two sets, but not in the intersection of their nearest common set family in the hierarchy.

Usage

```
outlying_elements(x, counts = TRUE)
```

Arguments

`x`	A HierarchicalSet object
`counts`	Should number of elements rather than the actual elements be returned. Defaults to `TRUE`

Value

A data.frame containing information on the outlying elements of each set pair. Only pairs with outlying elements are returned. The 'setX' coloumn contains the index of the first set in the pair and the 'setY' column contains the index of the second set in the pair. If `counts = TRUE` then the 'nOutliers' column contains the number of outlying elements for each pair. If `counts = FALSE` the the 'outlier' column contains the index of the outlying elements for each pair

See Also

`plot_outlier_distribution()` for plotting the distribution of outlying elements in a HierarchicalSet object

Examples

```
data('twitter')

twitSet <- create_hierarchy(twitter)

# Just get the counts
countOut <- outlying_elements(twitSet)
head(countOut)

# Or the actual elements
elemOut <- outlying_elements(twitSet, FALSE)
head(elemOut)
```

`plot.HierarchicalSet` *Visualize hierarchical sets*

Description

This is the main visualization interface to HierarchicalSet object. By changing the type argument you control which types of plots are produced. See datails for a walkthrough of the different plot types. All plots are based on ggplot2 but heavily modified using gtable. Because of this the return value is always a gtable object, so it is not possible to add additional geoms, or change scales etc. on the result of `plot()`.

Usage

```
## S3 method for class 'HierarchicalSet'
plot(
  x,
  label = TRUE,
  type = "dendrogram",
  transform = NULL,
  style = theme_bw(),
  quantiles = 0,
  upperBound = 1,
  tension = 0.8,
  alpha = 1,
  circular = TRUE,
  showHierarchy = !circular,
  evenHierarchy = circular,
  outliers = NULL,
  ratio = NULL,
  n = 50,
  ...
)
```

Arguments

`x`	A HierarchicalSet object to plot.
`label`	logical. Should sets be labeled.
`type`	The type of plot to produce. See detail. The name of the type may be abbreviated.
`transform`	A string giving the scale transformation or a `scales::trans()` object.
`style`	A ggplot2 theme to use as basis for the plot. Defaults to theme_bw().
`quantiles`	The quantiles to split outlying elements up in for outlying_elements plot. If length is above one a facetted plot will be produced.
`upperBound`	The upper quantile threshold to include. Defaults to 1 (i.e. everything is included)

tension	The tension used for the hierarchical edge bundles in outlying_elements plot. Defaults to 0.8
alpha	The alpha level for the edge bundles. Defaults to 1
circular	Logical. Should the hierarchical edge bundles be laid out in a circular layout.
showHierarchy	Logical. For intersectionStack plots, should a dendrogram mapping union sizes be drawn above the icicle plot. For outlying_elements plots should a dendrogram be plotted below (for circular) or to the left (for linear) of the edge bundles.
evenHierarchy	Logical. Should the heights of the dendrogram used for constructing the edge bundles be evened out.
outliers	A precomputed data.frame with outlier information, as returned by `outlying_elements()`.
ratio	Should outliers be plotted as a ratio instead of the raw number. If `NULL` the raw number is used, If `"min"` the raw number is divided by the number of elements in the smallest set of the pair, if `"max"` the largest set, and if `"mean"` the mean pair size.
n	The number of coordinates to calculate for each edge in the outlyingElements plot.
...	Currently ignored

Details

Currently 4 different plottypes are available:

dendrogram Plots a horizontal dendrogram with the x-value mapped to the intersection size divided by the union size. This plot very clearly shows the rise in heterogenity as more and more sets are joined, and clearly shows clusters of very similar sets.

intersectStack Plots a bottom-up icicleplot with height showing the size of the intersection. In essence this plot communicates the same type of information as a Venn-diagram, but in a scalable way and only showing the intersections along the hierarchy. Box color maps to the degree (number of sets) of the intersection making high-degree high-intersection as well as low-degree low-intersection boxes stand out.

heatmap Plots a traditional heatmap showing all 2-degree intersections. The sets are organized according to the hierarchy so the result should show a number of squares along the diagonal. If two very similar sets have been forced apart by the clustering, this will show up nicely as high value squares away from the diagonal.

composite Combines dendrogram, intersectStack and heatmap into a composite plot.

outlyingElements Plots intersects between two sets that are missing from the intersect of their shared top node as hierarchical edge bundles. It helps detect deviations from the global structure as defined by the hierarchcial clustering.

Value

A gtable object invisibly. This function is mainly called for the side effect of creating a plot.

plot_outlier_distribution *Plot the outlying elements of a HierarchicalSet object*

Description

This function creates a scatter plot showing each outlying element as a function of the number of sets it is present in and the number of times it is outlying.

Usage

```
plot_outlier_distribution(x, alpha = 0.3, outliers = NULL)
```

Arguments

x	A HierarchicalSet object
alpha	The transparancy of the dots
outliers	Precomputed outlying elements as returned from outlying_elements()

Value

This function is called for its side effects

See Also

outlying_elements() for extracting outlying element information from a HierarchicalSet object

Examples

```
data('twitter')

twitSet <- create_hierarchy(twitter)
plot_outlier_distribution(twitSet)
```

power_trans *Create a power transformation object*

Description

This function can be used to create a proper trans object that encapsulates a power transformation (x^n).

Usage

```
power_trans(n)
```

Arguments

n	The degree of the power transformation

Value

A trans object

---

twitter *Followers of 100 twitter users*

Description

This dataset captures the followers of 100 highly followed twitter users. The dataset is anonymized and based on the ego-Twitter network from Stanford Large Network Dataset Collection.

Usage

```
twitter
```

Format

A ngCMatrix with the sets (users) as columns and elements (followers) as rows

Source

https://snap.stanford.edu/data/

Index