Package 'GridGraph'

August 23, 2018

Title Network Layout as Vertices Constrained to a Grid

Version 1.0.1

Description Laying out large networks/graphs can be difficult when there are very many connections/edges. This package implements a grid-constrained layout algorithm which places nodes/vertices in different cells on a grid to maintain an optimal distance from each other. The cells of the grid can then be plotted using colour to highlight properties of the nodes. Connections are not drawn and thus do not obscure important properties of the network.

Depends R (>= 3.4.0)

License LGPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

R topics documented:

`grid.plot` *Plot grid-constrained layout of network vertices.*

Description

`grid.plot` Plotting function for network vertices laid out on a grid.

Usage

```
grid.plot(layout, colour.matrix = NULL)
```

Arguments

layout	A numerical matrix (as returned by layout.grid.centroid.attraction in which case the cells of the matrix will consist of indices referring to the columns of the distance matrix provided to layout.grid.centroid.attraction).
colour.matrix	A matrix representing colours in any of the R formats. This matrix must be the same size and dimensions as the layout matrix. Each element describes a colour for the vertex referred to by the corresponding element in layout. See example for details on how to use this.

Details

This function provides a convenient way to plot results of layout.grid.centroid.attraction.

Value

Invisibly returns a matrix of colours corresponding to the colour.matrix matrix, if provided, or the default colouring otherwise. Default colours are grey for vertices and white for empty cells (used to pad the matrix to a square grid).

Examples

```
## Not run:
require(GridGraph)
# Load example data and create distance matrix
data(hipExpr)
network = cor(hipExpr, method="pearson", use="pairwise.complete.obs")
distance.matrix = 1 - (network * (network >= 0))

# Perform layout
threshold = 0.5
grid.layout = layout.grid.centroid.attraction(distance.matrix, threshold,
iterations = 20)

# Select a vertex (one of the genes/columns from the example data) and
# find neighbours (other genes that correlate with this seed)
seed.vertex = match("ENSMUSG00000004891", rownames(distance.matrix))
threshold = 0.5
seed.neighbours = which(distance.matrix[seed.vertex, ] <= threshold)

# Create a martix of colours, background = grey, missing values = white,
# seed vertex = red, neighbour vertices = blue
grid.plot.colours = matrix(rep("#E0E0E0", length(grid.layout)),
ncol=ncol(grid.layout))
grid.plot.colours[is.na(grid.layout)] = "#FFFFFF"
grid.plot.colours[match(seed.neighbours, grid.layout)] = "#006699"
grid.plot.colours[match(seed.vertex, grid.layout)] = "#E03000"

# Plot the layout
grid.plot(grid.layout, grid.plot.colours)

## End(Not run)
```

GridGraph — *Grid-constrained layout of network vertices.*

Description

The GridGraph package provides the layout function `layout.grid.centroid.attraction` to lay out vertices as cells on a grid where proximity on the grid corresponds to edge weight between vertices, a plotting function `grid.plot` to plot the layout, as well as two helper functions `matrix.map` and `matrix.unmap` to map between row/column coordinates and vector indices. There is also an example dataset `hipExpr` provided.

Functions provided

`layout.grid.centroid.attraction`

`grid.plot`

`matrix.map`

`matrix.unmap`

Data provided

`hipExpr`

hipExpr — *Expression data for 1000 genes.*

Description

A dataset containing the expression of 1000 genes/transcripts measured in brain (hippocampus) tissue of different strains of mice. This example dataset contains 1000 selected genes to provide a dataset of manageable size for testing correlation and network functions.

Usage

```
hipExpr
```

Format

A data frame with 74 rows and 1000 variables:

columns Gene/transcript expression in arbitrary units (log2 transformed). Column names are ENSEMBL gene identifiers.

Details

Note: by being stripped of their context, the data for the 1000 genes in this dataset have lost their biological significance. This dataset is intended only for testing functions requiring a particular data structure.

Source

Details about the original dataset are provided at http://www.genenetwork.org/webqtl/main.py?FormID=sharinginfo&GN_AccessionId=112

`layout.grid.centroid.attraction`
Grid-constrained layout of network vertices.

Description

`layout.grid.centroid.attraction` Layout of network vertices by iterative convergence to the centroid of all neighbours.

Usage

```
layout.grid.centroid.attraction(distance.matrix, threshold,
  row.size.factor = 1, col.size.factor = 1, increment = 0.5,
  extend = NULL, iterations = 1)
```

Arguments

`distance.matrix`	A matrix of *distances* representing an undirected weighted network. Distances are smaller for more correlated vertices and can be generated by subtracting the Pearson correlation (or equivalent) from 1 and setting all values above 1 to equal 1.
`threshold`	A threshold below which vertices are considered to be connected by an edge.
`row.size.factor`	A factor indicating by how many rows the grid should be padded (see the details for an explanation).
`col.size.factor`	A factor indicating by how many columns the grid should be padded (see the details for an explanation).
`increment`	A value between 0 and 1 determining what proportion of the displacement to the centroid the new cell is placed. Larger values will speed up, but potentially destabilise, convergence. Default is 0.5 (half the displacement).
`extend`	An optional matrix representing the state of a layout (as returned by a previous iteration of `layout.grid.centroid.attraction`). If provided, this will be taken as the starting state of the current layout. In this way, additional iterations can be added to a layout or graphs can be made at different time points during layout to observe convergence as an animation. The default (NULL) will start with a new layout. It is important that the random seed is set beforehand using `set.seed`.
`iterations`	The number of iterations that should be performed.

Details

This function places each vertex onto a grid such that the Euclidean distance between any two cells is optimally close to the weight of the edge connecting the corresponding vertices. In each iteration of the layout, the centroid of the cells corresponding to all neighbor vertices is calculated and the new position of the vertex is moved towards this centroid (by a fraction of the distance determined by the `increment` parameter. Displaced cells are shuffled along a mapped direct path between target cell and source vertex. After several iterations, cells with converge to a layout where more highly-connected vertices are placed in more closely adjacent cells and unconnected (orphan) nodes are displaced to the edges.

Value

A matrix of indices corresponding to the rows in `distance.matrix` and NAs for non-vertex cells.

Examples

```
## Not run:
require(GridGraph)
# Trivial example using the "mtcars" data set.
distance.matrix <- 1 - cor(mtcars)
distance.matrix[which(distance.matrix > 1)] <- 1
grid.layout <- layout.grid.centroid.attraction(distance.matrix, 0.8, row.size.factor=4,
                                               col.size.factor=4,
                                               iterations=100)
image(t(apply(grid.layout > 0, 2, rev))) # Visualise

## End(Not run)

# See grid.plot for a more involved example.
```

`matrix.map` *Map matrix indices to coordinates.*

Description

`matrix.map` Maps the index of a matrix to the row and column coordinates, given the dimensions of the matrix.

Usage

```
matrix.map(n, dim.m)
```

Arguments

`n`	The index of a cell (or cells) in the matrix.
`dim.m`	A numeric vector of length 2 containing the dimesions (rows, columns) of the matrix (as returned by `dim`).

Details

This is intended as a helper function for the GridGraph package. It may be of use in other situations when working with matrices.

The function assumes the matrix is filled by column (the default behaviour in R, see matrix).

This function is vectorised, so that n can be a vector.

Value

A list of length 2 containing the named vectors 'row' and 'col' each of the same length as n.

---

matrix.unmap *Map matrix coordinates to indices.*

Description

matrix.unmap Maps the row and column coordinates to the index of a matrix, given the dimensions of the matrix.

Usage

```
matrix.unmap(row, col, dim.m)
```

Arguments

row	The row index of a cell (or cells) in a matrix.
col	The column index of a cell (or cells) in a matrix.
dim.m	A numeric vector of length 2 containing the dimesions (rows, columns) of the matrix (as returned by dim).

Details

This is intended as a helper function for the GridGraph package. It may be of use in other situations when working with matrices.

The function assumes the matrix is filled by column (the default behaviour in R, see matrix).

This function is vectorised so that row and col can be vectors. The parameters row and col must be the same length.

Value

A vector of the same length as 'row' and 'col' corresponding to the index of the cell(s) when the matrix is considered as a vector.

Index