Package 'autograph'

July 2, 2025

Title Automatic Plotting of Many Graphs

Version 0.1.2

Date 2025-06-27

Description Visual exploration and presentation of networks should not be difficult.
This package includes functions for plotting networks and network-
related metrics with sensible and pretty defaults.
It includes 'ggplot2'-based plot methods for many popular network package classes.
It also includes some novel layout algorithms, and options for straightforward, consistent themes.

URL https://stocnet.github.io/autograph/

BugReports https://github.com/stocnet/autograph/issues

License MIT + file LICENSE

Language en-GB

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 3.6.0), manynet

Imports cli, dplyr (>= 1.1.0), ggdendro, ggplot2, tidyr

Suggests testthat (>= 3.0.0), patchwork

Enhances RSiena

Config/Needs/build roxygen2, devtools

Config/Needs/check covr, lintr, spelling

Config/Needs/website pkgdown

Config/testthat/parallel true

Config/testthat/edition 3

NeedsCompilation no

Author James Hollway [cre, aut, ctb] (IHEID, ORCID:
<https://orcid.org/0000-0002-8361-9647>),
Henrique Sposito [ctb] (ORCID: <https://orcid.org/0000-0003-3420-6085>)

Maintainer James Hollway <james.hollway@graduateinstitute.ch>

Repository CRAN

Date/Publication 2025-07-02 15:30:06 UTC

Contents

`ag_call` *Consistent palette calls*

Description

These functions assist in calling particular parts of a theme's palette. For example, `ag_base()` will return the current theme's base or background color, and `ag_highlight()` will return the color used in that theme to highlight one or more nodes, lines, or such.

Usage

```
ag_base()

ag_highlight()

ag_positive()

ag_negative()

ag_qualitative(number)

ag_sequential(number)
```

```
ag_divergent(number)
```

Arguments

number	Integer of how many category colours to return.

Value

One or more hexcodes as strings.

---

layout_tbl_graph_layered
Layered layout

Description

Layered layout

Usage

```
layout_tbl_graph_layered(.data, center = NULL, circular = FALSE, times = 4)
```

Arguments

.data	Some {manynet} compatible network data.
center, circular	Extra parameters required for {tidygraph} compatibility.
times	Integer of sweeps that the algorithm will pass through. By default 4.

Value

Returns a table of coordinates.

Examples

```
ties <- data.frame(
  from = c("A", "A", "B", "C", "D", "F", "F", "E"),
  to   = c("B", "C", "D", "E", "E", "E", "G", "G"),
  stringsAsFactors = FALSE)

coords <- layout_tbl_graph_layered(ties, times = 6)
coords
```

layout_tbl_graph_matching *Matching layout*

Description

This layout works to position nodes opposite their matching nodes. See `manynet::to_matching()` for more details on the matching procedure.

Usage

```
layout_tbl_graph_matching(.data, center = NULL, circular = FALSE, times = 1)
```

Arguments

`.data`	Some {manynet} compatible network data.
`center, circular, times`	Extra parameters required for {tidygraph} compatibility.

Value

Returns a table of nodes' x and y coordinates.

made_earlier *Precooked results for demonstrating plotting*

Description

These are all pre-cooked results objects, saved here to save time in testing and demonstrating how autograph plots look.

Usage

```
data(res_migraph_reg)

data(res_migraph_test)

data(res_migraph_diff)

data(res_manynet_diff)

data(res_siena_gof)

data(res_siena_influence)
```

```
data(res_siena_selection)

data(res_monan_traces)

data(res_monan_gof)
```

Format

An object of class `netlm` of length 15.

An object of class `network_test` of length 9.

An object of class `diffs_model` (inherits from `data.frame`) with 20 rows and 11 columns.

An object of class `diff_model` (inherits from `tbl_df`, `tbl`, `data.frame`) with 4 rows and 10 columns.

An object of class `sienaGOF` of length 1.

An object of class `influenceTable` (inherits from `data.frame`) with 25 rows and 4 columns.

An object of class `selectionTable` (inherits from `data.frame`) with 25 rows and 4 columns.

An object of class `traces.monan` of length 3.

An object of class `gof.stats.monan` of length 2.

`map_measure` *Plotting logical marks Plotting numeric measures*

Description

These functions plot distributions for node, tie, and network measures, as defined in the {manynet} package.

Usage

```
## S3 method for class 'node_measure'
plot(x, type = c("h", "d"), ...)

## S3 method for class 'tie_measure'
plot(x, type = c("h", "d"), ...)

## S3 method for class 'network_measures'
plot(x, ...)
```

Arguments

`x`	An object of "node_measure", "tie_measure", or "network_measures" class.
`type`	For node and tie measures, whether the plot should be "h" a histogram or "d" a density plot. By default "h".
`...`	Other arguments to be passed on.

Value

`plot.node_measure()` and `plot.tie_measure()` returns a histogram and/or density plot of the distribution of the measure.

`plot.network_measures()` returns a plot of the measure traced over time.

Examples

```
plot(manynet::node_deg(ison_karateka))
plot(manynet::tie_betweenness(ison_karateka))
```

---

map_member *Plotting categorical memberships*

Description

This plotting method operates on "node_member" class objects from the {manynet} package, plotting the dendrogram of their membership.

Usage

```
## S3 method for class 'node_member'
plot(x, ...)

## S3 method for class 'matrix'
plot(x, ..., membership = NULL)
```

Arguments

`x`	An object of "node_member" class, for example as a result of running `manynet::node_in_community()`.
`...`	Other arguments to be passed on.
`membership`	A "node_member" membership vector.

Value

`plot.node_member()` returns a dendrogram, with labels colored to indicate the different clusters, and with the optimal cutpoint shown by a dashed highlight line.

`plot.matrix()` returns a plot of an adjacency or incidency matrix, potentially with the rows and columns reordered to illustrate an additional membership vector.

Examples

```
plot(manynet::node_in_walktrap(ison_southern_women, "e"))
plot(as_matrix(ison_adolescents),
  membership = node_in_walktrap(ison_adolescents, "e"))
plot(as_matrix(ison_southern_women),
  membership = node_in_walktrap(ison_southern_women, "e"))
```

map_motifs *Plotting tabular motifs*

Description

These functions will plot graphs of the motifs used in a vector of results of e.g. a triad census.

Usage

```
## S3 method for class 'node_motif'
plot(x, ...)

## S3 method for class 'network_motif'
plot(x, ...)
```

Arguments

x	An object of "node_motif" class, e.g. resulting from a call to `manynet::node_by_triad()`.
...	Other arguments to be passed on.

Value

`plot.node_motif()` returns a set of graphs that illustrate the motifs mentioned in the results from a node_motif function in {manynet}.

`plot.network_motif()` returns a set of graphs that illustrate the motifs mentioned in the results from a net_motif function in {manynet}.

map_themes *Many themes*

Description

This function enables all plots to be quickly, easily and consistently themed. This is achieved by setting a theme option that enables the appropriate palette to be used for all autograph-consistent plotting methods.

The following themes are currently available: default, bw, iheid, ethz, uzh, rug, unibe, crisp, neon, rainbow.

Usage

```
stocnet_theme(theme = NULL)
```

Arguments

theme	String naming a theme. By default "default". This string can be capitalised or not.

Value

This function sets the theme and palette(s) to be used across all stocnet packages. The palettes are written to options and held there.

Examples

```
stocnet_theme("default")
plot(manynet::node_degree(ison_karateka))
stocnet_theme("rug")
plot(manynet::node_degree(ison_karateka))
```

---

match_color *Matching colors across palettes*

Description

Sometimes particular colours are coded in certain ways to facilitate interpretation. For example, perhaps primary colours or traffic light colours are used to represent some discrete options. Yet institutional palettes vary in terms of which colours they have available. This function uses the Euclidean distance of colours in CIELAB space to those of a target palette to find the closes corresponding colours.

Usage

```
match_color(colors, pal)
```

Arguments

colors	One or more hexcodes to match with colors from the palette.
pal	Optionally, a vector of hexcodes representing a palette in which to find matches. By default, the current theme's qualitative palette is used.

Value

A vector of hexcodes the length of the first argument.

Examples

```
match_color("#4575b4")
```

model_mrqap *Plotting methods for MRQAP models*

Description

These plotting methods are for results obtained by fitting an MRQAP model. The S3 classes are "netlm" or "netlogit", and so are compatible with the results from either the {sna} or {migraph} packages.

Usage

```
## S3 method for class 'netlm'
plot(x, ...)

## S3 method for class 'netlogit'
plot(x, ...)
```

Arguments

x	An object obtained by fitting an MRQAP model to some data. For example, migraph::net_regression().
...	Further arguments to be passed on to plot.

Value

A plot showing the location of observed statistics compared to the distribution of statistics from permuted networks.

Examples

```
# Here's something I cooked up with migraph earlier:
plot(res_migraph_reg)
```

plot.diffusion *Plotting diffusion models*

Description

Plotting diffusion models

Usage

```
## S3 method for class 'diff_model'
plot(x, ..., all_steps = TRUE)

## S3 method for class 'diffs_model'
plot(x, ...)

## S3 method for class 'learn_model'
plot(x, ...)
```

Arguments

x	A "diff_model" of "diffs_model" class of object. E.g. as a result from `manynet::play_diffusion()`.
...	Other arguments to be passed.
all_steps	Whether all steps should be plotted or just those where there is change in the distributions.

Value

`plot.diff_model()` returns a bar chart of the number of new infected nodes at each time point, as well as an overlay line plot of the total of infected

Examples

```
plot(res_manynet_diff)
plot(res_migraph_diff)
plot(play_learning(ison_networkers, beliefs = runif(net_nodes(ison_networkers))))
```

---

plot.influenceTable *Plotting influence tables*

Description

These are functions for constructing and presenting influence tables for the interpretation of results for network and behavior dynamics obtained with the RSiena or multiSiena packages.

Usage

```
## S3 method for class 'influenceTable'
plot(x, separation = 0, ...)
```

Arguments

x	An object of class "influenceTable", created using `RSiena::influenceTable()`.
separation	This can be used to make the curves visually distinguishable if they overlap too much without it. An advisable value then is, e.g., 0.01.
...	Other arguments to be passed.

Value

A plot showing how the influence evaluation function changes based on ego's value and alter's value of some covariate.

Author(s)

Tom Snijders

References

Consult also the RSiena manual, Sections 13.2 and 13.4. Gratitude to Steffen Triebel and Rene Veenstra for corrections.

Examples

```
plot(res_siena_influence)
```

---

plot.network_test *Plotting methods for CUG and QAP tests*

Description

These plotting methods are for results obtained by testing some statistic against those produced in a reference distribution of conditional uniform graphs or as a quadratic assignment procedure. The S3 class is "network_test".

Usage

```
## S3 method for class 'network_test'
plot(x, ..., threshold = 0.95, tails = c("two", "one"))
```

Arguments

x	An object obtained from a conditional uniform graph or quadratic assignment procedure test. For example, `migraph::test_permutation()`.
...	Other arguments to be passed on.
threshold	The empirical threshold to shade in the plot.
tails	By default "two" indicating a two-tailed test, but "one" for a one-tailed test is also available.

Value

A distribution of the simulated or permuted statistics, with 2.5% shaded at each end, and a line highlighting where the observed statistic lies on this distribution.

Examples

```
# Here's something I cooked up with migraph earlier:
plot(res_migraph_test)
```

`plot.selectionTable` *Plotting selection tables*

Description

These are functions for constructing and presenting selection tables for the interpretation of results for network dynamics obtained with the RSiena package.

Usage

```
## S3 method for class 'selectionTable'
plot(x, quad = TRUE, separation = 0, ...)
```

Arguments

`x`	An object of class "selectionTable", created using `RSiena::selectionTable()`.
`quad`	When TRUE (the default), a quadratic function (average and total alter) is plotted. Use `quad = FALSE` for similarity effects.
`separation`	This can be used to make the curves visually distinguishable if they overlap too much without it. An advisable value then is, e.g., 0.01.
`...`	Other arguments to be passed.

Value

A plot showing how the selection evaluation function changes based on ego's value and alter's value of some covariate.

Author(s)

Tom Snijders

References

Consult also the RSiena manual, Sections 13.1 and 13.3.

Examples

```
plot(res_siena_selection)
```

plot.sienaGOF *SIENA Goodness of Fit*

Description

This function plots goodness of fit objects created using RSiena. Unlike the plot method included in the {RSiena} package, this function utilises {ggplot2} and not {lattice}, which makes the output more compatible and themeable.

Usage

```
## S3 method for class 'sienaGOF'
plot(x, ...)
```

Arguments

x	A sienaGOF object, as returned by RSiena::sienaGOF().
...	Other parameters to be passed to the plotting funciton, for example main = "Title" for a different title than the default.

Value

A violin plot showing the distribution of statistics from the simulations and a line highlighting the observed statistics.

Examples

```
plot(res_siena_gof)
```

plot_monan_gof *plot.gof.stats.monan*

Description

plot.gof.stats.monan

Usage

```
## S3 method for class 'gof.stats.monan'
plot(x, lvls, ...)
```

Arguments

x	An object of class "gof.stats.monan".
lvls	The values for which the gofFunction should be calculated/plotted.
...	Additional plotting parameters, use discouraged.

Value

The function `plot.gof.stats.monan` returns violin plots of the gof tests with observed values superimposed in red.

Examples

```
plot(res_monan_gof, lvls = 1:15)
```

`plot_monan_trace` *plot.traces.monan*

Description

plot.traces.monan

Usage

```
## S3 method for class 'traces.monan'
plot(x, ...)
```

Arguments

`x`	An object of class "traces.monan".
`...`	Additional plotting parameters, use not recommended.

Value

The function `plot.traces.monan` shows a scatter plot of the statistics of simulated networks from phase three of the esimtation.

Examples

```
plot(res_monan_traces)
```

Index