Package 'causalHyperGraph'

July 22, 2025

Type Package

Title Drawing Causal Hypergraphs

Version 0.1.0

Date 2024-03-18

Author Christoph Falk [aut, cph],
Mathias Ambuehl [aut, cre],
Michael Baumgartner [aut]

Imports DiagrammeR, DiagrammeRsvg, stringr, useful, cna, utils

Maintainer Mathias Ambuehl <mathias.ambuehl@consultag.ch>

Description Draws causal hypergraph plots from models output by configurational comparative methods such as Coincidence Analysis (CNA) or Qualitative Comparative Analysis (QCA).

Encoding UTF-8

License GPL (>= 2)

RoxygenNote 7.3.1

NeedsCompilation no

Repository CRAN

Date/Publication 2024-03-21 17:40:05 UTC

Contents

---

causalHyperGraph   *Draw Causal Hypergraphs*

Description

causalHyperGraph() draws causal hypergraphs based on solution formulas of configurational comparative methods as **cna** or **QCA**. In contrast to a directed acyclic graph (DAG), edges of a hypergraph can connect more than two nodes. chg() is a short form/alias of causalHyperGraph().

Usage

```
causalHyperGraph(
  x,
  show_formula = FALSE,
  formula_font = "Courier",
  formula_spaces = NULL,
  n = 10,
  ask = length(x) > 1
)

chg(
  x,
  show_formula = FALSE,
  formula_font = "Courier",
  formula_spaces = NULL,
  n = 10,
  ask = length(x) > 1
)

## S3 method for class 'causalHyperGraph'
plot(x, n = attr(x, "n"), ask = attr(x, "ask"), print = TRUE, ...)

## S3 method for class 'causalHyperGraph'
print(x, n = attr(x, "n"), ask = attr(x, "ask"), plot = TRUE, ...)

## S3 method for class 'causalHyperGraph'
x[i, ask = length(out) > 1]

## S3 method for class 'causalHyperGraph'
c(..., n = 10, ask = length(out) > 1)
```

Arguments

x	A character vector containing atomic or complex solution formulas (asf or csf) in crisp-set (binary) or multi-value standard form.
show_formula	Logical; if TRUE, the formula is printed below the graph.

`formula_font`	Character string; specifies a font for the formula. The name of any available systemfont can be used. The argument is only relevant if `show_formula=TRUE`.
`formula_spaces`	Character vector; the characters in this vector will be displayed with a space around them in the formula. The argument is only relevant if `show_formula=TRUE`.
`n`	Positive integer; specifies the maximal number of graphs to render.
`ask`	Logical; if `TRUE`, the user is asked to hit <Return> before a new graph is drawn.
`print`	Logical; if `TRUE`, `x` is printed to the console.
`...`	Arguments passed to methods.
`plot`	Logical; if `TRUE`, `x` is rendered graphically (in addition to being printed to the console).
`i`	A vector (integer, character or logical) indicating elements to select.

Details

The most common type of graph representing causal structures is the directed acyclic graph (DAG) (cf. e.g. Spirtes et al. 2000). Edges in DAGs connect exactly two nodes, and the edges indicate the direction of causation. In contrast, edges in hypergraphs can connect more than two nodes and, thereby, represent more than just the direction of causation. Causal hypergraphs represent causal complexity, that is, they depict the fact that often many causes need to be instantiated jointly, i.e. as a bundle, in order for an effect to occur (which is known as *causal conjunctivity*) and that different bundles of causes can bring about an effect independently of one another on alternative causal routes (*causal disjunctivity*). Hypergraphs are particularly useful for representing causal structures in the vein of Mackie's (1974) INUS theory of causation, and modern variants thereof (Baumgartner & Falk 2023).

The methods designed to trace causal complexity in data are configurational comparative methods (CCMs), such as **cna** (Baumgartner & Ambühl 2020) and **QCA** (Ragin 2008). Accordingly, the first argument of `causalHyperGraph()` is a character vector of CCM solution formulas in crisp-set (binary) or multi-value standard form (asf or csf). CCM solution formulas are conjunctions of biconditionals with minimized disjunctive normal forms (DNF) on the left-hand sides and single effects on the right-hand sides. Conjunction is expressed by "∗", disjunction by "+", negation by changing upper case into lower case letters and vice versa, conditional by "→", and biconditional by "↔". Examples are `(A*b + c*B <-> D)*(D + E <-> F)` or `(A=2*B=4 + A=3*B=1 <-> C=2)*(C=2*D=3 + C=1*D=4 <-> E=3)`.

If the hypergraphs drawn by `causalHyperGraph()` have crisp-set (or fuzzy-set) factors only, upper case "X" means X=1, lower case "x" stands for X=0, and "◇" expresses that the value of the factor at the tail of the edge is negated. In case of hypergraphs with multi-value factors, the relevant values of the factors are displayed at the tails and heads of the directed edges. In all graphs, nodes whose exiting edges are joined by "●" form a conjunction, and the tails of edges with the same head node (so-called "colliders") constitute a disjunction.

The arguments `show_formula`, `formula_font`, and `formula_spaces` control the display of the solution formula below the graph. `show_formula` determines whether the formula is printed, and `formula_font` specifies a font for the formula. The argument `formula_spaces` identifies characters that are displayed with a space around them. For example, `formula_spaces = c("+", "<->")` displays "+" and "↔" with a space around them. `formula_font` and `formula_spaces` only have an effect if `show_formula` is set to its non-default value `TRUE`.

The argument n specifies the maximal number of graphs to render. If the number of graphs is larger than n, only the first n graphs are drawn. By means of the argument ask the rendering of the graphs can be paused. If ask=TRUE, the user is asked to hit <Return> before a new graph is drawn. If ask=FALSE, all n graphs are drawn at once.

Formally, causalHyperGraph() returns a list of graphs of class "causalHyperGraph" produced using the **DiagrammeR** package. Such a list contains one or more graphs.

The class "causalHyperGraph" has the following methods: plot() for rendering the graphs, print() for printing the solution formulas to the console and, optionally, graph rendering, c() for concatenating several "causalHyperGraph" objects, and []/subset() for subsetting. By contrast, extraction of a single list element with [[]] or $ does not return anything useful.

Hint: Use length(x) to query the number of graphs in an object of class "causalHyperGraph".

Value

causalHyperGraph() returns a list of class "causalHyperGraph" containing one or several graphs.

References

Baumgartner, Michael and Christoph Falk. 2023. Boolean Difference-Making: A Modern Regularity Theory of Causation. *The British Journal for the Philosophy of Science* 74(1):171–197.

Baumgartner, Michael and Mathias Ambühl. 2020. Causal Modeling with Multi-Value and Fuzzy-Set Coincidence Analysis. *Political Science Research and Methods* 8:526–542.

Mackie, John L. 1974. *The Cement of the Universe: A Study of Causation.* Oxford: Oxford University Press.

Ragin, Charles C. 2008. *Redesigning Social Inquiry: Fuzzy Sets and Beyond.* Chicago: University of Chicago Press.

Spirtes, Peter, Clark Glymour and Richard Scheines. 2000. *Causation, Prediction, and Search*. 2 ed. Cambridge: MIT Press.

See Also

export_as_svg, and the methods described in plot.cna.

Examples

```
library(cna)  # required for randomAsf(), randomCsf(), and allCombs()

x <- "(A+B<->C)*(B+D<->E)"
causalHyperGraph(x)
chg(x)

# Input of length > 1
x <- c("(A*b+a*B<->C)*(C+f<->E)", "(A*B+a*b<->C)*(C+F<->E)")
gr1 <- causalHyperGraph(x)
gr1
# Suppress plotting.
print(gr1, plot = FALSE)
```

```
# Outcomes can be negated.
chg("(A+B<->c)*(c+F<->E)")

# Negative outcomes that appear positively downstream are rendered
# with double negation in the resulting hypergraph.
chg("(A+B<->c)*(C+F<->E)")

# Random formula.
x <- randomCsf(6, n.asf = 2)
chg(x, show_formula = TRUE)
# Change the font of the formula.
chg(x, show_formula = TRUE, formula_font = "arial")
# Change the spacing.
chg(x, show_formula = TRUE, formula_spaces = c("+", "<->"))

# Multi-value formula.
x <- "(C=1*G=0 + T=1*A=0 + T=2*G=3 <-> P=1)*(P=1*M=0 + F=1 <-> D=1)"
causalHyperGraph(x)

# Random multi-value formula with 3 outcomes.
x <- randomCsf(allCombs(c(3,3,3,3,3,3)), n.asf = 3)
gr2 <- causalHyperGraph(x)
gr2

# Random multi-value formula with a random number of outcomes.
y <- randomCsf(allCombs(c(3,4,3,5,3,4)))
gr3 <- chg(y, show_formula = TRUE)
gr3

# Concatenation.
gr4 <- c(gr1,gr2,gr3)
plot(gr4)

# Subsetting.
gr5 <- gr4[c(1,4)]
plot(gr5)

# Longer factor names.
x <- paste("(Test1=1*Test2=3+Test3=2 <-> Out1=2)",
           "(Out1=1*TestN=5 <-> Out2=3)",
           "(TestN=4+TestK=1*Out2=1 <-> Out3=5)",
           sep = "*")
chg(x)
```

---

export_as_svg *Export causal hypergraphs in svg format*

Description

export_as_svg() exports the hypergraphs to svg files for editing with svg software.

Usage

```
export_as_svg(x, file, sep = "", verbose = TRUE)
```

Arguments

x	A graph generated with causalHyperGraph().
file	Character vector specifying file names for the graphs in x, optionally with a path. The file names must not include the extension ".svg", which will be appended automatically. file can be of length 1 or length(x). If length(file)<length(x), the name(s) in file is (are) recycled to length(x) and adjusted with make.unique().
sep	Separator passed to make.unique().
verbose	Logical; if TRUE (default), additional information about the file names is printed in the console. If FALSE, no information at all is printed.

Details

Although causalHyperGraph() optimizes the placement of the nodes and edges (using **DiagrammeR**) as much as possible, the resulting graph may still need manual adjustments to enhance its readability. In case of multi-value graphs, for example, factor values may be packed too closely or even printed on top of each other, calling for manual disentangling. Once the graphs have been exported via export_as_svg() such adjustments can be made with any svg software, such as Inkscape, Adobe Illustrator, or Apache Batik. Moreover, svg files can easily be converted to pdf or other formats.

Value

Returns the names of the generated file(s) invisibly.

See Also

causalHyperGraph, and the methods plot.cna.

Examples

```
library(cna)  # required for randomCsf() and allCombs()

# Manual crisp-set example.
x <- "(X+Y<->S)*(D+s*B<->E)*(A+S*B<->C)*(C+E<->F)"
gr1 <- causalHyperGraph(x)
export_as_svg(gr1, file.path(tempdir(), "csGraph"))

# Random multi-value example.
y <- randomCsf(allCombs(c(3,4,3,5,3,4)))
gr2 <- chg(y)
export_as_svg(gr2, file.path(tempdir(), "mvGraph"))

file.remove(file.path(tempdir(), c("csGraph.svg", "mvGraph.svg")))
```

---

plot.cna	*Create hypergraphs from solution objects of* cna() *and related functions*

Description

These plot() methods draw causal hypergraphs from the solution formulas included in the objects of the following classes from package **cna**:

- class "cna", the output of cna();
- class "condTbl", the output of condTbl(), csf(), asf() and msc();
- class "condList", the output of condition().

Usage

```
## S3 method for class 'condTbl'
plot(x, outcome = NULL, n = 10, ask = nrow(x) > 1, ...)

## S3 method for class 'cna'
plot(x, what = c("csf", "asf", "msc"), ...)

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

Arguments

x	An object of an appropriate class.
outcome	Character vector of factor names. Only solutions with one of these factors as outcome will be plotted. If NULL (default), all solutions are plotted.
n	Positive integer; specifies the maximal number of graphs to render.
ask	Logical; if TRUE, the user is asked to hit <Return> before a new graph is drawn.
...	Additional arguments in plot.condTbl() are passed to causalHyperGraph(), in particular show_formula. Additional arguments in plot.cna() and plot.condition() are passed to plot.condTbl(), e.g. n, outcome, ask.
what	One of "csf" (default), "asf", "msc", determining which type of solution to extract from the "cna" object.

Value

These plot() methods return an object of class "causalHyperGraph".

Examples

```
library(cna)

ana <- cna(d.educate)
pl <- plot(asf(ana))
pl

plot(csf(ana))
plot(csf(ana), ask = FALSE)

plot(ana)
plot(ana, show_formula = TRUE)
plot(ana, what = "msc", n = 5)

plot(msc(ana), outcome = "E")
plot(ana, what = "msc", outcome = "E")  # same as previous
```

Index