

# Package ‘bnviewer’

October 12, 2022

**Type** Package

**Title** Bayesian Networks Interactive Visualization and Explainable Artificial Intelligence

**Version** 0.1.6

**Depends** R (>= 3.4)

**Description** Bayesian networks provide an intuitive framework for probabilistic reasoning and its graphical nature can be interpreted quite clearly. Graph based methods of machine learning are becoming more popular because they offer a richer model of knowledge that can be understood by a human in a graphical format. The 'bnviewer' is an R Package that allows the interactive visualization of Bayesian Networks. The aim of this package is to improve the Bayesian Networks visualization over the basic and static views offered by existing packages.

**License** MIT + file LICENSE

**URL** <http://robsonfernandes.net/bnviewer/>

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Repository** CRAN

**Imports** visNetwork (>= 2.0.4), bnlearn (>= 4.3), methods, igraph (>= 1.2.4), shiny, assertthat, caret, e1071, graphics

**NeedsCompilation** no

**Author** Robson Fernandes [aut, cre, cph]

**Maintainer** Robson Fernandes <robson.fernandes@usp.br>

**Date/Publication** 2020-09-14 06:00:03 UTC

## R topics documented:

bnviewer-package . . . . .	2
bn.to.igraph . . . . .	2
model.to.structure . . . . .	3
strength.viewer . . . . .	4
viewer . . . . .	9

**Index****14**

---

bnviewer-package      *Interactive Visualization of Bayesian Networks*

---

**Description**

Bayesian networks provide an intuitive framework for probabilistic reasoning and its graphical nature can be interpreted quite clearly. Graph based methods of machine learning are becoming more popular because they offer a richer model of knowledge that can be understood by a human in a graphical format. The 'bnviewer' is an R Package that allows the interactive visualization of Bayesian Networks. The aim of this package is to improve the Bayesian Networks visualization over the basic and static views offered by existing packages.

**Details**

Package: bnviewer  
Type: Package  
Version: 0.1.5  
Date: 2020-07-22  
License: MIT + file LICENSE

**Author(s)**

Robson Fernandes  
Institute of Mathematical and Computer Sciences  
University of Sao Paulo - ICMC-USP  
Maintainer: Robson Fernandes <robson.fernandes@usp.br>

---

bn.to.igraph      *Bayesian Network to iGraph Model*

---

**Description**

Converts Bayesian network structure based on package "bnlearn" and "bnviewer" to model based on package "igraph".

**Usage**

```
bn.to.igraph(bayesian.network)
```



```

model.right.arrow.op2 = " A => (B,C,D);
                        B => (E,F);
                        F => (G);
                        "

structure = model.to.structure(model.left.arrow.op1)

viewer(structure,
        bayesianNetwork.width = "100%",
        bayesianNetwork.height = "80vh",
        bayesianNetwork.layout = "layout_on_grid",
        node.colors = list(background = "#f4bafd",
                            border = "#2b7ce9",
                            highlight = list(background = "#97c2fc",
                                                border = "#2b7ce9"))
        )

data.set = bnlearn::gaussian.test

bayesianNetwork.fit = bn.fit(structure, data = data.set)

print(bayesianNetwork.fit$A)

```

---

strength.viewer

*Interactive Bayesian Network Strength Viewer*


---

## Description

Show the strength of the probabilistic relationships expressed by the arcs of a Bayesian network, and use model averaging to build a network containing only the significant arcs.

## Usage

```

strength.viewer(
  bayesianNetwork,
  bayesianNetwork.boot.strength,
  bayesianNetwork.background = NULL,
  bayesianNetwork.arc.strength.threshold.expression = NULL,
  bayesianNetwork.arc.strength.threshold.expression.color = NULL,
  bayesianNetwork.arc.strength.threshold.alternative.color = NULL,
  bayesianNetwork.arc.strength.label = FALSE,
  bayesianNetwork.arc.strength.label.prefix = "",
  bayesianNetwork.arc.strength.label.color = NULL,
  bayesianNetwork.arc.strength.tooltip = FALSE,
  bayesianNetwork.edge.scale.min = 1,
  bayesianNetwork.edge.scale.max = 5,
  bayesianNetwork.edge.scale.label.min = 14,

```

```

    bayesianNetwork.edge.scale.label.max = 14,
    bayesianNetwork.title = "",
    bayesianNetwork.subtitle = "",
    bayesianNetwork.footer = "",
    bayesianNetwork.enabled.interactive.mode = FALSE,
    bayesianNetwork.layout = "default",
    bayesianNetwork.width = "100%",
    bayesianNetwork.height = "500px",
    node.shape = NULL,
    node.label.prefix = "",
    node.colors = list(),
    node.font = list(),
    node.size = 10,
    edges.smooth = TRUE,
    edges.dashes = FALSE,
    edges.colors = list(),
    edges.width = 1,
    options.highlightNearest = TRUE,
    options.nodesIdSelection = FALSE,
    clusters.legend.title = "",
    clusters.legend.options = list(),
    clusters = list()
)

```

## Arguments

```

bayerianNetwork
    A Bayesian Network structure from Averaged Network
bayerianNetwork.boot.strength
    A nonparametric bootstrap to assess arc strength and direction
bayerianNetwork.background
    Bayesian network background
bayerianNetwork.arc.strength.threshold.expression
    Logical expression of the force threshold of the arcs of the Bayesian network
bayerianNetwork.arc.strength.threshold.expression.color
    Color applied to logical expression of the force threshold of the arcs of the
    Bayesian network
bayerianNetwork.arc.strength.threshold.alternative.color
    Alternative color to logical expression of the force threshold of the arcs of the
    Bayesian network
bayerianNetwork.arc.strength.label
    Enable Bayesian Network arc strength label
bayerianNetwork.arc.strength.label.prefix
    Include Bayesian Network arc strength label prefix
bayerianNetwork.arc.strength.label.color
    Set Bayesian Network arc strength label color
bayerianNetwork.arc.strength.tooltip
    Enable Bayesian Network arc strength tooltip

```

bayesianNetwork.edge.scale.min  
Set bayesian Network edge scale minimum

bayesianNetwork.edge.scale.max  
Set bayesian Network edge scale maximum

bayesianNetwork.edge.scale.label.min  
Set bayesian Network edge scale label minimum

bayesianNetwork.edge.scale.label.max  
Set bayesian Network edge scale label maximum

bayesianNetwork.title  
: String. Bayesian Network title

bayesianNetwork.subtitle  
: String. Bayesian Network subtitle

bayesianNetwork.footer  
: String. Bayesian Network footer

bayesianNetwork.enabled.interactive.mode  
: Boolean. Enabled interactive viewer mode.

bayesianNetwork.layout  
: String. A layout of a Bayesian Network

1. layout\_on\_sphere
2. layout\_on\_grid
3. layout\_in\_circle
4. layout\_as\_star
5. layout\_as\_tree
6. layout\_with\_sugiyama
7. layout\_with\_kk
8. layout\_with\_dh
9. layout\_with\_lgl
10. layout\_with\_mds
11. layout\_with\_gem
12. layout\_nicely
13. layout\_components
14. layout\_hierarchical\_direction\_UD
15. layout\_hierarchical\_direction\_DU
16. layout\_hierarchical\_direction\_LR
17. layout\_hierarchical\_direction\_RL

bayesianNetwork.width  
: String. Bayesian Network width

bayesianNetwork.height  
: String. Bayesian Network height

node.shape : String. A node shape of a Bayesian Network

1. dot (default)
2. circle
3. ellipse

	4. database
	5. diamond
	6. square
	7. triangle
	8. box
	9. star
	10. text
node.label.prefix	: String. Adds a prefix to the node label
node.colors	: String   named list. Color for the node. Can be 'rgba(120,32,14,1)', '#97C2FC' (hexa notation on 7 char without transparency) or 'red'. Can be just one color, or a list with several elements: <ol style="list-style-type: none"> <li>1. "background" : String. Default to '#97C2FC'. Background color for the node.</li> <li>2. "border" : String. Default to '#2B7CE9'. Border color for the node.</li> <li>3. "highlight" : String   named list, Color of the node when selected. <ol style="list-style-type: none"> <li>(a) "background" : String. Default to '#97C2FC'. Background color for the node when selected.</li> <li>(b) "border" : String. Default to '#2B7CE9'. Border color for the node when selected.</li> </ol> </li> </ol>
node.font	Node Font : Array. Example list(color = "black", face="Arial")
node.size	Integer. Node Size.
edges.smooth	: Boolean. When true, the edge is drawn as a dynamic quadratic bezier curve.
edges.dashes	: Array or Boolean. Default to false. When true, the edge will be drawn as a dashed line.
edges.colors	: Named list or String. Default to named list. Color information of the edge in every situation. Can be 'rgba(120,32,14,1)', '#97C2FC' (hexa notation on 7 char without transparency) or 'red'. <ul style="list-style-type: none"> <li>• "color" : String. Default to '#848484'. The color of the edge when it is not selected or hovered over (assuming hover is enabled in the interaction module).</li> <li>• "highlight " : String. Default to '#848484'. The color the edge when it is selected.</li> <li>• "hover" : String. Default to '#848484'. The color the edge when the mouse hovers over it (assuming hover is enabled in the interaction module).</li> <li>• "inherit" : String or Boolean. Default to 'from'. When color, highlight or hover are defined, inherit is set to false! Supported options are: true, false, 'from', 'to', 'both'.</li> <li>• "opacity" : Number. Default to 1.0. It can be useful to set the opacity of an edge without manually changing all the colors. The allowed range of the opacity option is between 0 and 1.</li> </ul>
edges.width	: Number. Default to 1. Sets edge width.
options.highlightNearest	: Boolean. Default to true. Highlight nearest when clicking a node.

```

options.nodesIdSelection
    : Boolean. Default to false. Add an id node selection creating an HTML select
      element.
clusters.legend.title
    : Array. Get details in the example.
clusters.legend.options
    : Array of Array. Get details in the example.
clusters
    : Array of Array. Get details in the example.

```

## References

See online documentation <http://robsonfernandes.net/bnviewer>

## Examples

```

library(bnlearn)
library(bnviewer)

bayesianNetwork.boot.strength = boot.strength(coronary, R = 20, algorithm = "hc")

avg.bayesianNetwork = averaged.network(bayesianNetwork.boot.strength, threshold = 0.2)

strength.viewer(
  avg.bayesianNetwork,
  bayesianNetwork.boot.strength,
  bayesianNetwork.background = "white",
  bayesianNetwork.arc.strength.threshold.expression = c("@threshold > 0 & @threshold < 0.5",
                                                         "@threshold >= 0.5 & @threshold < 0.6",
                                                         "@threshold >= 0.6 & @threshold <= 1"),

  bayesianNetwork.arc.strength.threshold.expression.color = c("red", "yellow", "green"),
  bayesianNetwork.arc.strength.threshold.alternative.color = "white",

  bayesianNetwork.arc.strength.label = TRUE,
  bayesianNetwork.arc.strength.label.prefix = "",
  bayesianNetwork.arc.strength.label.color = "black",

  bayesianNetwork.arc.strength.tooltip = TRUE,

  bayesianNetwork.edge.scale.min = 1,
  bayesianNetwork.edge.scale.max = 3,

  bayesianNetwork.edge.scale.label.min = 14,
  bayesianNetwork.edge.scale.label.max = 14,

  bayesianNetwork.width = "100%",
  bayesianNetwork.height = "800px",
  bayesianNetwork.layout = "layout_with_sugiyama",
  node.colors = list(background = "#97c2fc",
                     border = "#2b7ce9",
                     highlight = list(background = "#e91eba",

```



```

border = "#2b7ce9")),

node.font = list(color = "black", face="Arial"),
edges.dashes = FALSE,

bayesianNetwork.title="Bayesian Network Strength Analysis - Coronary",
bayesianNetwork.subtitle = "Coronary heart disease data set",
bayesianNetwork.footer = "Fig. 1 - Layout with Sugiyama"
)

```

---

viewer

*Interactive Bayesian Network Viewer*


---

## Description

Interactive Bayesian Network Viewer

## Usage

```

viewer(
  bayesianNetwork,
  bayesianNetwork.background = NULL,
  bayesianNetwork.title = "",
  bayesianNetwork.subtitle = "",
  bayesianNetwork.footer = "",
  bayesianNetwork.enabled.interactive.mode = FALSE,
  bayesianNetwork.layout = "default",
  bayesianNetwork.width = "100%",
  bayesianNetwork.height = "500px",
  bayesianNetwork.data = NULL,
  bayesianNetwork.correlation.show = FALSE,
  bayesianNetwork.correlation.method = "pearson",
  node.shape = NULL,
  node.label.prefix = "",
  node.colors = list(),
  node.font = list(),
  edges.smooth = TRUE,
  edges.dashes = FALSE,
  edges.width = 1,
  edges.font.size = 18,
  edges.font.color = "black",
  edges.shadow = FALSE,
  options.highlightNearest = TRUE,
  options.nodesIdSelection = FALSE,
  clusters.legend.title = "",
  clusters.legend.options = list(),
  clusters = list()
)

```

**Arguments**

`bayesianNetwork`  
A Bayesian Network structure. (Example : hill-climbing (HC)).

`bayesianNetwork.background`  
Bayesian network background

`bayesianNetwork.title`  
: String. Bayesian Network title

`bayesianNetwork.subtitle`  
: String. Bayesian Network subtitle

`bayesianNetwork.footer`  
: String. Bayesian Network footer

`bayesianNetwork.enabled.interactive.mode`  
: Boolean. Enabled interactive viewer mode.

`bayesianNetwork.layout`  
: String. A layout of a Bayesian Network. The hierarchical layout the available options are: UD, DU, LR, RL. To simplify: up-down, down-up, left-right, right-left.

1. `layout_on_sphere`
2. `layout_on_grid`
3. `layout_in_circle`
4. `layout_as_star`
5. `layout_as_tree`
6. `layout_with_sugiyama`
7. `layout_with_kk`
8. `layout_with_dh`
9. `layout_with_lgl`
10. `layout_with_mds`
11. `layout_with_gem`
12. `layout_nicely`
13. `layout_components`
14. `layout_hierarchical_direction_UD`
15. `layout_hierarchical_direction_DU`
16. `layout_hierarchical_direction_LR`
17. `layout_hierarchical_direction_RL`

`bayesianNetwork.width`  
: String. Bayesian Network width

`bayesianNetwork.height`  
: String. Bayesian Network height

`bayesianNetwork.data`  
: List. Data Set.

`bayesianNetwork.correlation.show`  
: Boolean. When true, show Correlation Coefficient. Default (FALSE).

`bayesianNetwork.correlation.method`  
 : String. Correlation Coefficient. One of "pearson" (default), "kendall", or "spearman"

`node.shape`  
 : String. A node shape of a Bayesian Network
 

1. dot (default)
2. circle
3. ellipse
4. database
5. diamond
6. square
7. triangle
8. box
9. star
10. text

`node.label.prefix`  
 : String. Adds a prefix to the node label

`node.colors`  
 : String | named list. Color for the node. Can be 'rgba(120,32,14,1)', '#97C2FC' (hexa notation on 7 char without transparency) or 'red'. Can be just one color, or a list with several elements:
 

1. "background" : String. Default to '#97C2FC'. Background color for the node.
2. "border" : String. Default to '#2B7CE9'. Border color for the node.
3. "highlight" : String | named list, Color of the node when selected.
  - (a) "background" : String. Default to '#97C2FC'. Background color for the node when selected.
  - (b) "border" : String. Default to '#2B7CE9'. Border color for the node when selected.

`node.font`  
 : Array. Example list(color = "black", face="Arial")

`edges.smooth`  
 : Boolean. When true, the edge is drawn as a dynamic quadratic bezier curve.

`edges.dashes`  
 : Array or Boolean. Default to false. When true, the edge will be drawn as a dashed line.

`edges.width`  
 : Number. Sets edge width. Default to 1.

`edges.font.size`  
 : Number. Font Size. Default to 18.

`edges.font.color`  
 : String. Font Color. Default to 'black'.

`edges.shadow`  
 : Boolean. Shadow. Default to FALSE.

`options.highlightNearest`  
 : Boolean. Default to true. Highlight nearest when clicking a node.

`options.nodesIdSelection`  
 : Boolean. Default to false. Add an id node selection creating an HTML select element.

```
clusters.legend.title
    : Array. Get details in the example.
clusters.legend.options
    : Array of Array. Get details in the example.
clusters
    : Array of Array. Get details in the example.
```

## References

See online documentation <http://robsonfernandes.net/bnviewer>

See the code fontAwesome for icons in groups and nodes <https://fontawesome.com/v4.7.0/cheatsheet/>

## Examples

```
library(bnlearn)
library(bnviewer)

data("alarm")
bayesianNetwork = hc(alarm)

viewer(bayesianNetwork,
  bayesianNetwork.background = "-webkit-radial-gradient(center, ellipse cover,
                                rgba(255,255,255,1) 0%,
                                rgba(246,246,246,1) 47%,
                                rgba(237,237,237,1) 100%)",

  bayesianNetwork.width = "100%",
  bayesianNetwork.height = "100vh",
  bayesianNetwork.layout = "layout_components",
  bayesianNetwork.title = "<br>Discrete Bayesian Network - Alarm",
  bayesianNetwork.subtitle = "Monitoring of emergency care patients",

  node.colors = list(background = "white",
                     border = "black",
                     highlight = list(background = "#e91eba",
                                       border = "black")),
  node.font = list(color = "black", face="Arial"),

  clusters.legend.title = list(text = "<b>Legend</b> <br> Variable Categories",
                              style = "font-size:18px;
                                       font-family:Arial;
                                       color:black;
                                       text-align:center;"),

  clusters.legend.options = list(
    list(label = "Pressure",
         shape = "icon",
         icon = list(code = "f1ce",
                    size = 50,
                    color = "#e91e63")),
    list(label = "Volume",
         shape = "icon",
```

```

        icon = list(code = "f140",
                    size = 50,
                    color = "#03a9f4"),
list(label = "Ventilation",
      shape = "icon",
      icon = list(code = "f192",
                  size = 50,
                  color = "#4caf50")),
list(label = "Saturation",
      shape = "icon",
      icon = list(code = "f10c",
                  size = 50,
                  color = "#ffc107"))
),

clusters = list(
  list(label = "Pressure",
        shape = "icon",
        icon = list(code = "f1ce", color = "#e91e63"),
        nodes = list("CVP", "BP", "HRBP", "PAP", "PRSS")),
  list(label = "Volume",
        shape = "icon",
        icon = list(code = "f140", color = "#03a9f4"),
        nodes = list("MINV", "MVS", "LVV", "STKV")),
  list(label = "Ventilation",
        shape = "icon",
        icon = list(code = "f192", color = "#4caf50"),
        nodes = list("VALV", "VLNG", "VTUB", "VMCH")),
  list(label = "Saturation",
        shape = "icon",
        icon = list(code = "f10c", color = "#ffc107"),
        nodes = list("HRSA", "SA02", "PVS"))
)
)

```

# Index

`bn.to.igraph`, [2](#)  
`bnviewer` (`bnviewer-package`), [2](#)  
`bnviewer-package`, [2](#)  
`model.to.structure`, [3](#)  
`strength.viewer`, [4](#)  
`viewer`, [9](#)