

# Package ‘CBNplot’

April 29, 2026

**Type** Package

**Title** plot bayesian network inferred from gene expression data based on enrichment analysis results

**Version** 1.13.0

**Description** This package provides the visualization of bayesian network inferred from gene expression data. The networks are based on enrichment analysis results inferred from packages including clusterProfiler and ReactomePA. The networks between pathways and genes inside the pathways can be inferred and visualized.

**License** Artistic-2.0

**Encoding** UTF-8

**Depends** R (>= 4.3.0)

**Imports** ggplot2, magrittr, graphite, ggraph, igraph, bnlearn (>= 4.7), patchwork, org.Hs.eg.db, clusterProfiler, utils, enrichplot, reshape2, ggforce, dplyr, tidyr, stringr, depmap, ExperimentHub, Rmpfr, graphlayouts, BiocFileCache, ggdist, purrr, pvelust, stats, rlang

**Suggests** knitr, arules, concaveman, ReactomePA, bnviewer, rmarkdown, withr, BiocStyle, testthat (>= 3.0.0)

**biocViews** Visualization, Bayesian, GeneExpression, NetworkInference, Pathways, Reactome, Network, NetworkEnrichment, GeneSetEnrichment

**VignetteBuilder** knitr

**RoxygenNote** 7.3.2

**URL** <https://github.com/noriakis/CBNplot>

**BugReports** <https://github.com/noriakis/CBNplot/issues>

**Config/testthat/edition** 3

**git\_url** <https://git.bioconductor.org/packages/CBNplot>

**git\_branch** devel

**git\_last\_commit** f259146

**git\_last\_commit\_date** 2026-04-28

**Repository** Bioconductor 3.24

**Date/Publication** 2026-04-29

**Author** Noriaki Sato [cre, aut]

**Maintainer** Noriaki Sato <nori@hgc.jp>

## Contents

|                            |           |
|----------------------------|-----------|
| CBNplot-package . . . . .  | 2         |
| bngeneplot . . . . .       | 3         |
| bngeneplotCustom . . . . . | 6         |
| bngenetest . . . . .       | 9         |
| bnpathplot . . . . .       | 10        |
| bnpathplotCustom . . . . . | 12        |
| bnpathtest . . . . .       | 15        |
| compareBNs . . . . .       | 17        |
| exampleEaRes . . . . .     | 17        |
| exampleGeneExp . . . . .   | 18        |
| inferMS . . . . .          | 18        |
| loadSign . . . . .         | 19        |
| obtainPath . . . . .       | 19        |
| queryCpDistLs . . . . .    | 20        |
| queryCpDistLw . . . . .    | 20        |
| <b>Index</b>               | <b>22</b> |

---

|                 |   |
|-----------------|---|
| CBNplot-package | <i>CBNplot: plot bayesian network inferred from gene expression data based on enrichment analysis results</i> |
|-----------------|---|

---

## Description

This package provides the visualization of bayesian network inferred from gene expression data. The networks are based on enrichment analysis results inferred from packages including clusterProfiler and ReactomePA. The networks between pathways and genes inside the pathways can be inferred and visualized.

## Details

[CBNplot::bngeneplot()] - The main function using gene expression data within the pathway to infer Bayesian network. [CBNplot::bnpathplot()] - The main function using pathway expression data, which is defined by eigengene of the gene expression values to infer Bayesian network. [CBNplot::bngeneplotCustom()] - The function that provides custom visualization for the gene network. [CBNplot::bnpathplotCustom()] - The function that provides custom visualization for the pathway network.

## Author(s)

**Maintainer:** Noriaki Sato <nori@hgc.jp>

## See Also

Useful links:

- <https://github.com/noriakis/CBNplot>
- Report bugs at <https://github.com/noriakis/CBNplot/issues>

---

**bngeneplot***bngeneplot*

---

**Description**

Plot gene relationship within the specified pathway

**Usage**

```
bngeneplot(  
  results,  
  exp,  
  expSample = NULL,  
  algo = "hc",  
  R = 20,  
  returnNet = FALSE,  
  algorithm.args = NULL,  
  bypassConverting = FALSE,  
  edgeLink = FALSE,  
  pathNum = NULL,  
  convertSymbol = TRUE,  
  expRow = "ENSEMBL",  
  interactive = FALSE,  
  cexCategory = 1,  
  cl = NULL,  
  showDir = FALSE,  
  chooseDir = FALSE,  
  scoreType = "bic-g",  
  labelSize = 4,  
  layout = "nicely",  
  clusterAlpha = 0.2,  
  strType = "normal",  
  delZeroDegree = TRUE,  
  otherVar = NULL,  
  otherVarName = NULL,  
  onlyDf = FALSE,  
  disc = FALSE,  
  tr = NULL,  
  remainCont = NULL,  
  sp = "hsapiens",  
  compareRef = FALSE,  
  compareRefType = "intersection",  
  pathDb = "reactome",  
  dep = NULL,  
  depMeta = NULL,  
  sizeDep = FALSE,  
  showDepHist = TRUE,  
  cellLineName = "5637_URINARY_TRACT",  
  showLineage = FALSE,  
  orgDb = org.Hs.eg.db,  
  shadowText = TRUE,  
)
```

```

    bgColor = "white",
    textColor = "black",
    strengthPlot = FALSE,
    nStrength = 10,
    strThresh = NULL,
    hub = NULL,
    seed = 1,
    useSiGN = FALSE
)

```

### Arguments

|                  |  |
|------------------|--|
| results          | the enrichment analysis result   |
| exp              | gene expression matrix   |
| expSample        | candidate samples to be included in the inference default to all   |
| algo             | structure learning method used in boot.strength() default to "hc"  |
| R                | the number of bootstrap  |
| returnNet        | whether to return the network  |
| algorithm.args   | parameters to pass to bnlearn structure learning function  |
| bypassConverting | bypass the symbol converting If you use custom annotation databases that does not have SYMBOL listed in keys. ID of rownames and those listed in EA result must be same. |
| edgeLink         | use geom_edge_link() instead of geom_edge_diagonal()   |
| pathNum          | the pathway number (the number of row of the original result, ordered by p-value)  |
| convertSymbol    | whether the label of resulting network is converted to symbol, default to TRUE   |
| expRow           | the type of the identifier of rows of expression matrix  |
| interactive      | whether to use bnviewer (default to FALSE)   |
| cexCategory      | scaling factor of size of nodes  |
| cl               | cluster object from parallel::makeCluster()  |
| showDir          | show the confidence of direction of edges  |
| chooseDir        | if undirected edges are present, choose direction of edges (default: FALSE)  |
| scoreType        | score type to use on choosing direction  |
| labelSize        | the size of label of the nodes   |
| layout           | ggraph layout, default to "nicely"   |
| clusterAlpha     | if specified multiple pathways, the parameter is passed to geom_mark_hull()  |
| strType          | "normal" or "ms" for multiscale implementation   |
| delZeroDegree    | delete zero degree nodes   |
| otherVar         | other variables to be included in the inference  |
| otherVarName     | the names of other variables   |
| onlyDf           | return only data.frame used for inference  |
| disc             | discretize the expressoin data   |
| tr               | Specify data.frame if one needs to discretize as the same parametersas the other dataset   |

|                |   |
|----------------|---|
| remainCont     | Specify characters when perform discretization, if some columns are to be remain continuous   |
| sp             | query to <code>graphite::pathways()</code> , default to "hsapiens"  |
| compareRef     | whether compare to the reference network  |
| compareRefType | "intersection" or "difference"  |
| pathDb         | query to <code>graphite::pathways()</code> , default to "reactome"  |
| dep            | the tibble storing dependency score from library <code>depmap</code>  |
| depMeta        | <code>depmap::depmap_metadata()</code> , needed for <code>showLineage</code>  |
| sizeDep        | whether to reflect <code>DepMap</code> score to the node size   |
| showDepHist    | whether to show <code>depmap</code> histogram   |
| cellLineName   | the cell line name to be included   |
| showLineage    | show the dependency score across the lineage  |
| orgDb          | perform <code>clusterProfiler::setReadable</code> based on this organism database   |
| shadowText     | whether to use shadow text for the better readability default: TRUE   |
| bgColor        | color for text background when <code>shadowText</code> is TRUE  |
| textColor      | color for text when <code>shadowText</code> is TRUE   |
| strengthPlot   | append the barplot depicting edges with high strength   |
| nStrength      | specify how many edges are included in the strength plot  |
| strThresh      | the threshold for strength  |
| hub            | visualize the genes with top-n hub scores   |
| seed           | A random seed to make the analysis reproducible, default is 1.  |
| useSiGN        | default to FALSE. For using SiGN-BN in the function in Windows 10/11, 1. Download the SiGN-BN HC+BS binary in WSL ( <a href="https://sign.hgc.jp/signbn/download.html">https://sign.hgc.jp/signbn/download.html</a> )<br>2. Set PATH to executable (sign.1.8.3) |

**Value**

ggplot2 object

**Examples**

```
data("exampleEaRes");data("exampleGeneExp")
res <- bngeneplot(results = exampleEaRes, exp = exampleGeneExp, pathNum = 1,
  R = 10, convertSymbol = TRUE, expRow = "ENSEMBL")
```

---

|                  |                         |
|------------------|-------------------------|
| bngeneplotCustom | <i>bngeneplotCustom</i> |
|------------------|-------------------------|

---

## Description

Plot gene relationship within the specified pathway using customized theme

## Usage

```
bngeneplotCustom(  
  results,  
  exp,  
  expSample = NULL,  
  algo = "hc",  
  R = 20,  
  pathNum = NULL,  
  convertSymbol = TRUE,  
  expRow = "ENSEMBL",  
  interactive = FALSE,  
  cexCategory = 1,  
  cl = NULL,  
  showDir = FALSE,  
  chooseDir = FALSE,  
  algorithm.args = NULL,  
  labelSize = 4,  
  layout = "nicely",  
  strType = "normal",  
  returnNet = FALSE,  
  otherVar = NULL,  
  otherVarName = NULL,  
  onlyDf = FALSE,  
  disc = FALSE,  
  tr = NULL,  
  remainCont = NULL,  
  dep = NULL,  
  sizeDep = FALSE,  
  orgDb = org.Hs.eg.db,  
  bypassConverting = FALSE,  
  edgeLink = FALSE,  
  cellLineName = "5637_URINARY_TRACT",  
  fontFamily = "sans",  
  strengthPlot = FALSE,  
  nStrength = 10,  
  strThresh = NULL,  
  hub = NULL,  
  glowEdgeNum = NULL,  
  nodePal = c("blue", "red"),  
  edgePal = c("blue", "red"),  
  textCol = "black",  
  titleCol = "black",  
  backCol = "white",
```

```

    barTextCol = "black",
    barPal = c("red", "blue"),
    barBackCol = "white",
    scoreType = "bic-g",
    barLegendKeyCol = "white",
    barAxisCol = "black",
    bg.colour = NULL,
    bg.r = 0.1,
    barPanelGridCol = "black",
    titleSize = 24,
    seed = 1
)

```

### Arguments

|                |   |
|----------------|---|
| results        | the enrichment analysis result  |
| exp            | gene expression matrix  |
| expSample      | candidate rows to be included in the inference default to all                               |
| algo           | structure learning method used in boot.strength() default to "hc"                           |
| R              | the number of bootstrap   |
| pathNum        | the pathway number (the number of row of the original result, ordered by p-value)           |
| convertSymbol  | whether the label of resulting network is converted to symbol, default to TRUE              |
| expRow         | the type of the identifier of rows of expression matrix                                     |
| interactive    | whether to use bnviewer (default to FALSE)  |
| cexCategory    | scaling factor of size of nodes   |
| cl             | cluster object from parallel::makeCluster()   |
| showDir        | show the confidence of direction of edges   |
| chooseDir      | if undirected edges are present, choose direction of edges                                  |
| algorithm.args | parameters to pass to bnlearn structure learning function                                   |
| labelSize      | the size of label of the nodes  |
| layout         | ggraph layout, default to "nicely"  |
| strType        | "normal" or "ms" for multiscale implementation  |
| returnNet      | whether to return the network   |
| otherVar       | other variables to be included in the inference   |
| otherVarName   | the names of other variables  |
| onlyDf         | return only data.frame used for inference   |
| disc           | discretize the expressoin data  |
| tr             | Specify data.frame if one needs to discretize as the same parameters as the other dataset   |
| remainCont     | Specify characters when perform discretization, if some columns are to be remain continuous |
| dep            | the tibble storing dependency score from library depmap                                     |
| sizeDep        | whether to reflect DepMap score to the node size  |
| orgDb          | perform clusterProfiler::setReadable based on this organism database                        |

|                               |  |
|-------------------------------|--|
| <code>bypassConverting</code> | bypass the symbol converting ID of rownames and those listed in EA result must be same |
| <code>edgeLink</code>         | use <code>geom_edge_link()</code> instead of <code>geom_edge_diagonal()</code>         |
| <code>cellLineName</code>     | the cell line name to be included  |
| <code>fontFamily</code>       | font family name to be used for plotting   |
| <code>strengthPlot</code>     | append the barplot depicting edges with high strength                                  |
| <code>nStrength</code>        | specify how many edges are included in the strength plot                               |
| <code>strThresh</code>        | the threshold for strength   |
| <code>hub</code>              | visualize the genes with top-n hub scores  |
| <code>glowEdgeNum</code>      | edges with top-n confidence of direction are highlighted                               |
| <code>nodePal</code>          | vector of coloring of nodes (low, high)  |
| <code>edgePal</code>          | vector of coloring of edges (low, high)  |
| <code>textCol</code>          | color of texts in network plot   |
| <code>titleCol</code>         | color of title in network plot   |
| <code>backCol</code>          | color of background in network plot  |
| <code>barTextCol</code>       | text color in barplot  |
| <code>barPal</code>           | bar color  |
| <code>barBackCol</code>       | background color in barplot  |
| <code>scoreType</code>        | score type to use on inference   |
| <code>barLegendKeyCol</code>  | legend key color in barplot  |
| <code>barAxisCol</code>       | axis color in barplot  |
| <code>bg.colour</code>        | parameter to pass to <code>geom_node_text</code>                                       |
| <code>bg.r</code>             | parameter to pass to <code>geom_node_text</code>                                       |
| <code>barPanelGridCol</code>  | panel grid color in barplot  |
| <code>titleSize</code>        | the size of title  |
| <code>seed</code>             | A random seed to make the analysis reproducible, default is 1.                         |

**Value**

ggplot2 object

**Examples**

```
data("exampleEaRes");data("exampleGeneExp")
res <- bngeneplotCustom(results=exampleEaRes, exp=exampleGeneExp,
  pathNum=1, glowEdgeNum=NULL, hub=3, R=40,
  fontFamily="sans")
```

---

bngenetest

*bngenetest*


---

### Description

Testing various R for bayesian network between genes

### Usage

```
bngenetest(
  results,
  exp,
  expSample = NULL,
  algo = "hc",
  Rrange = seq(2, 40, 2),
  cl = NULL,
  algorithm.args = NULL,
  pathNum = NULL,
  convertSymbol = TRUE,
  expRow = "ENSEMBL",
  scoreType = "aic-g",
  orgDb = org.Hs.eg.db,
  bypassConverting = FALSE
)
```

### Arguments

|                  |   |
|------------------|---|
| results          | the enrichment analysis result  |
| exp              | gene expression matrix  |
| expSample        | candidate rows to be included in the inference default to all                     |
| algo             | structure learning method used in boot.strength() default to "hc"                 |
| Rrange           | the sequence of R values to be tested   |
| cl               | cluster object from parallel::makeCluster()                                       |
| algorithm.args   | parameters to pass to bnlearn structure learning function                         |
| pathNum          | the pathway number (the number of row of the original result, ordered by p-value) |
| convertSymbol    | whether the label of resulting network is converted to symbol, default to TRUE    |
| expRow           | the type of the identifier of rows of expression matrix                           |
| scoreType        | return the specified scores   |
| orgDb            | perform clusterProfiler::setReadable based on this organism database              |
| bypassConverting | bypass symbol converting  |

### Value

list of graphs and scores

## Examples

```
data("exampleEaRes");data("exampleGeneExp")
res <- bngenetest(results = exampleEaRes, exp = exampleGeneExp,
  algo="hc", Rrange=seq(10, 30, 10), pathNum=1, scoreType="bge")
```

---

bnpathplot

*bnpathplot*

---

## Description

Plot pathway relationship

## Usage

```
bnpathplot(
  results,
  exp,
  expSample = NULL,
  algo = "hc",
  algorithm.args = NULL,
  expRow = "ENSEMBL",
  cl = NULL,
  returnNet = FALSE,
  otherVar = NULL,
  otherVarName = NULL,
  qvalueCutOff = NULL,
  adjpCutOff = 0.05,
  nCategory = 15,
  R = 20,
  interactive = FALSE,
  color = "p.adjust",
  cexCategory = 1,
  cexLine = 0.5,
  chooseDir = FALSE,
  showDir = FALSE,
  delZeroDegree = TRUE,
  labelSize = 4,
  layout = "nicely",
  onlyDf = FALSE,
  disc = FALSE,
  tr = NULL,
  remainCont = NULL,
  shadowText = TRUE,
  bgColor = "white",
  textColor = "black",
  compareRef = FALSE,
  strThresh = NULL,
  strType = "normal",
  hub = NULL,
  scoreType = "bic-g",
  databasePal = "Set2",
```

```

dep = NULL,
sizeDep = FALSE,
orgDb = org.Hs.eg.db,
bypassConverting = FALSE,
useSiGN = FALSE,
edgeLink = TRUE,
cellLineName = "5637_URINARY_TRACT",
strengthPlot = FALSE,
nStrength = 10,
seed = 1
)

```

### Arguments

|                |   |
|----------------|---|
| results        | the enrichment analysis result  |
| exp            | gene expression matrix  |
| expSample      | candidate rows to be included in the inference default to all                               |
| algo           | structure learning method used in boot.strength() default to "hc"                           |
| algorithm.args | parameters to pass to bnlearn structure learning function                                   |
| expRow         | the type of the identifier of rows of expression matrix                                     |
| cl             | cluster object from parallel::makeCluster()   |
| returnNet      | whether to return the network   |
| otherVar       | other variables to be included in the inference   |
| otherVarName   | the names of other variables  |
| qvalueCutOff   | the cutoff value for qvalue   |
| adjpCutOff     | the cutoff value for adjusted pvalues   |
| nCategory      | the number of pathways to be included   |
| R              | the number of bootstrap   |
| interactive    | whether to use bnviewer (default to FALSE)  |
| color          | color of node, default to adjusted p-value  |
| cexCategory    | scaling factor of size of nodes   |
| cexLine        | scaling factor of width of edges  |
| chooseDir      | if undirected edges are present, choose direction of edges                                  |
| showDir        | show the confidence of direction of edges   |
| delZeroDegree  | delete zero degree nodes  |
| labelSize      | the size of label of the nodes  |
| layout         | ggraph layout, default to "nicely"  |
| onlyDf         | return only data.frame used for inference   |
| disc           | discretize the expressoin data  |
| tr             | Specify data.frame if one needs to discretize as the same parameters as the other dataset   |
| remainCont     | Specify characters when perform discretization, if some columns are to be remain continuous |
| shadowText     | whether to use shadow text for the better readability (default: TRUE)                       |

|                               |   |
|-------------------------------|---|
| <code>bgColor</code>          | color for text background when <code>shadowText</code> is TRUE  |
| <code>textColor</code>        | color for text when <code>shadowText</code> is TRUE   |
| <code>compareRef</code>       | whether compare to the reference network between pathway  |
| <code>strThresh</code>        | threshold for strength, automatically determined if NULL  |
| <code>strType</code>          | "normal" or "ms" for multiscale implementation  |
| <code>hub</code>              | change the shape of node according to hub scores (default NULL)   |
| <code>scoreType</code>        | score type to use on choosing edge direction  |
| <code>databasePal</code>      | palette to be used in <code>scale_color_brewer</code> when the multiple results are to be shown   |
| <code>dep</code>              | the tibble storing dependency score from library <code>depmap</code>  |
| <code>sizeDep</code>          | whether to reflect <code>DepMap</code> score to the node size   |
| <code>orgDb</code>            | perform <code>clusterProfiler::setReadable</code> based on this organism database   |
| <code>bypassConverting</code> | bypass the symbol converting If you use custom annotation databases that does not have SYMBOL listed in keys. ID of rownames and those listed in EA result must be same.  |
| <code>useSiGN</code>          | default to FALSE. For using SiGN-BN in the function in Windows 10/11, 1. Download the SiGN-BN HC+BS binary in WSL ( <a href="https://sign.hgc.jp/signbn/download.html">https://sign.hgc.jp/signbn/download.html</a> )<br>2. Set PATH to executable (sign.1.8.3) |
| <code>edgeLink</code>         | whether to set edge to <code>geom_edge_link()</code> FALSE to use <code>geom_edge_diagonal()</code>   |
| <code>cellLineName</code>     | the cell line name to be included   |
| <code>strengthPlot</code>     | append the barplot depicting edges with high strength   |
| <code>nStrength</code>        | specify how many edges are included in the strength plot  |
| <code>seed</code>             | A random seed to make the analysis reproducible, default is 1.  |

**Value**

`ggplot2` object

**Examples**

```
data("exampleEaRes");data("exampleGeneExp")
res <- bnpathplot(results = exampleEaRes, exp = exampleGeneExp,
  R = 10, expRow = "ENSEMBL")
```

---

`bnpathplotCustom`

*bnpathplotCustom*

---

**Description**

Plot pathway relationship using customized theme

**Usage**

```
bnpathplotCustom(  
  results,  
  exp,  
  expSample = NULL,  
  algo = "hc",  
  R = 20,  
  expRow = "ENSEMBL",  
  color = "p.adjust",  
  cexCategory = 1,  
  cl = NULL,  
  showDir = FALSE,  
  chooseDir = FALSE,  
  labelSize = 4,  
  layout = "nicely",  
  strType = "normal",  
  compareRef = FALSE,  
  disc = FALSE,  
  tr = NULL,  
  remainCont = NULL,  
  qvalueCutOff = NULL,  
  adjpCutOff = 0.05,  
  nCategory = 15,  
  cexLine = 1,  
  returnNet = FALSE,  
  dep = NULL,  
  sizeDep = FALSE,  
  cellLineName = "5637_URINARY_TRACT",  
  fontFamily = "sans",  
  otherVar = NULL,  
  otherVarName = NULL,  
  onlyDf = FALSE,  
  algorithm.args = NULL,  
  strengthPlot = FALSE,  
  nStrength = 10,  
  edgeLink = FALSE,  
  strThresh = NULL,  
  hub = NULL,  
  glowEdgeNum = NULL,  
  nodePal = c("blue", "red"),  
  edgePal = c("blue", "red"),  
  textCol = "black",  
  backCol = "white",  
  barTextCol = "black",  
  barPal = c("red", "blue"),  
  barBackCol = "white",  
  scoreType = "bic-g",  
  barLegendKeyCol = "white",  
  orgDb = org.Hs.eg.db,  
  barAxisCol = "black",  
  barPanelGridCol = "black",  
  bg.colour = NULL,  
)
```

```

    bg.r = 0.1,
    seed = 1,
    bypassConverting = FALSE
  )

```

### Arguments

|                |   |
|----------------|---|
| results        | the enrichment analysis result  |
| exp            | gene expression matrix  |
| expSample      | candidate rows to be included in the inference default to all                               |
| algo           | structure learning method used in boot.strength() default to "hc"                           |
| R              | the number of bootstrap   |
| expRow         | the type of the identifier of rows of expression matrix                                     |
| color          | color of node, default to adjusted p-value  |
| cexCategory    | scaling factor of size of nodes   |
| cl             | cluster object from parallel::makeCluster()   |
| showDir        | show the confidence of direction of edges   |
| chooseDir      | if undirected edges are present, choose direction of edges                                  |
| labelSize      | the size of label of the nodes  |
| layout         | ggraph layout, default to "nicely"  |
| strType        | "normal" or "ms" for multiscale implementation  |
| compareRef     | whether compare to the reference network between pathway                                    |
| disc           | discretize the expressoin data  |
| tr             | Specify data.frame if one needs to discretize as the same parameters as the other dataset   |
| remainCont     | Specify characters when perform discretization, if some columns are to be remain continuous |
| qvalueCutOff   | the cutoff value for qvalue   |
| adjpCutOff     | the cutoff value for adjusted pvalues   |
| nCategory      | the number of pathways to be included   |
| cexLine        | scaling factor of width of edges  |
| returnNet      | whether to return the network   |
| dep            | the tibble storing dependency score from library depmap                                     |
| sizeDep        | whether to reflect DepMap score to the node size  |
| cellLineName   | the cell line name to be included   |
| fontFamily     | font family name to be used for plotting  |
| otherVar       | other variables to be included in the inference   |
| otherVarName   | the names of other variables  |
| onlyDf         | return only data.frame used for inference   |
| algorithm.args | parameters to pass to bnlearn structure learning function                                   |
| strengthPlot   | append the barplot depicting edges with high strength                                       |
| nStrength      | specify how many edges are included in the strength plot                                    |
| edgeLink       | use geom_edge_link() instead of geom_edge_diagonal()  |

|                  |  |
|------------------|--|
| strThresh        | threshold for strength, automatically determined if NULL                               |
| hub              | change the shape of node according to hub scores (default NULL)                        |
| glowEdgeNum      | edges with top-n confidence of direction are highlighted                               |
| nodePal          | vector of coloring of nodes (low, high)  |
| edgePal          | vector of coloring of edges (low, high)  |
| textCol          | color of texts in network plot   |
| backCol          | color of background in network plot  |
| barTextCol       | text color in barplot  |
| barPal           | bar color  |
| barBackCol       | background color in barplot  |
| scoreType        | score type to use on inference   |
| barLegendKeyCol  | legend key color in barplot  |
| orgDb            | perform clusterProfiler::setReadable based on this organism database                   |
| barAxisCol       | axis color in barplot  |
| barPanelGridCol  | panel grid color in barplot  |
| bg.colour        | parameter to pass to geom_node_text  |
| bg.r             | parameter to pass to geom_node_text  |
| seed             | A random seed to make the analysis reproducible, default is 1.                         |
| bypassConverting | bypass the symbol converting ID of rownames and those listed in EA result must be same |

**Value**

ggplot2 object

**Examples**

```
data("exampleEaRes");data("exampleGeneExp")
res <- bnpathplotCustom(results=exampleEaRes, exp=exampleGeneExp,
                        fontFamily="sans", glowEdgeNum=3, hub=3)
```

---

 bnpathtest

*bnpathtest*


---

**Description**

Testing various R for bayesian network between pathways

**Usage**

```

bnpathtest(
  results,
  exp,
  expSample = NULL,
  algo = "hc",
  algorithm.args = NULL,
  expRow = "ENSEMBL",
  cl = NULL,
  orgDb = org.Hs.eg.db,
  bypassConverting = FALSE,
  qvalueCutOff = 0.05,
  adjpCutOff = 0.05,
  nCategory = 15,
  Rrange = seq(2, 40, 2),
  scoreType = "aic-g"
)

```

**Arguments**

|                  |  |
|------------------|--|
| results          | the enrichment analysis result                                       |
| exp              | gene expression matrix   |
| expSample        | candidate rows to be included in the inference default to all        |
| algo             | structure learning method used in boot.strength() default to "hc"    |
| algorithm.args   | parameters to pass to bnlearn structure learning function            |
| expRow           | the type of the identifier of rows of expression matrix              |
| cl               | cluster object from parallel::makeCluster()                          |
| orgDb            | perform clusterProfiler::setReadable based on this organism database |
| bypassConverting | bypass symbol converting   |
| qvalueCutOff     | the cutoff value for qvalue  |
| adjpCutOff       | the cutoff value for adjusted pvalues                                |
| nCategory        | the number of pathways to be included                                |
| Rrange           | the sequence of R values to be tested                                |
| scoreType        | return the specified scores  |

**Value**

list of graphs and scores

**Examples**

```

data("exampleEaRes");data("exampleGeneExp")
res <- bnpathtest(results = exampleEaRes, exp = exampleGeneExp,
  algo="hc", Rrange=seq(10, 30, 10), expRow = "ENSEMBL",
  scoreType="bge")

```

---

`compareBNs`*compareBNs*

---

**Description**

Take the list of networks and returns the F-measures

**Usage**

```
compareBNs(listOfNets)
```

**Arguments**

`listOfNets` list of networks

**Value**

F-measures of each combination of network

**Examples**

```
data("exampleEaRes");data("exampleGeneExp")
net1 <- bngeneplot(results = exampleEaRes,
  exp = exampleGeneExp, pathNum = 1, R = 10, returnNet=TRUE)
net2 <- bngeneplot(results = exampleEaRes,
  exp = exampleGeneExp, pathNum = 1, R = 10, returnNet=TRUE)
res <- compareBNs(list(net1$av, net2$av))
```

---

`exampleEaRes`*Example enrichment analysis result*

---

**Description**

An example enrichment analysis result to be used for testing purpose. The result was produced by running `ReactomePA::enrichPathway()` and subsequent `clusterProfiler::setReadable()` on 'exampleGeneExp'.

**Usage**

```
data(exampleEaRes)
```

**Format**

An object of class `enrichResult` with 47 rows and 9 columns.

**Value**

example enrichment analysis result

---

|                |                                     |
|----------------|-------------------------------------|
| exampleGeneExp | <i>Example gene expression data</i> |
|----------------|-------------------------------------|

---

**Description**

An example gene expression data to be used for testing purpose made by runif() for ERCC genes and 100 samples. No biological meanings can be obtained from the data.

**Usage**

```
data(exampleGeneExp)
```

**Format**

An object of class `data.frame` with 7 rows and 100 columns.

**Value**

example gene expression

---

|         |                |
|---------|----------------|
| inferMS | <i>inferMS</i> |
|---------|----------------|

---

**Description**

multiscale bootstrap-based inference of Bayesian network

**Usage**

```
inferMS(data, algo, algorithm.args, R, cl = NULL, r = seq(0.5, 1.5, 0.1))
```

**Arguments**

|                             |  |
|-----------------------------|--|
| <code>data</code>           | data.frame to perform inference                                |
| <code>algo</code>           | structure learning method used in <code>boot.strength()</code> |
| <code>algorithm.args</code> | parameters to pass to bnlearn structure learning function      |
| <code>R</code>              | the number of bootstrap  |
| <code>cl</code>             | cluster object from <code>parallel::makeCluster()</code>       |
| <code>r</code>              | vector for size of each bootstrap replicate                    |

**Value**

object of class `bn.strength`

---

|          |                 |
|----------|-----------------|
| loadSign | <i>loadSign</i> |
|----------|-----------------|

---

**Description**

Load the output of SiGN-BN (HC+BS)

**Usage**

```
loadSign(fileName)
```

**Arguments**

|          |                       |
|----------|-----------------------|
| fileName | the result of SiGN-BN |
|----------|-----------------------|

**Value**

list of edges, nodes, strength, and bn (bnlearn)

---

|            |                   |
|------------|-------------------|
| obtainPath | <i>obtainPath</i> |
|------------|-------------------|

---

**Description**

obtain the analysis results including the queried gene symbol

**Usage**

```
obtainPath(res, geneSymbol)
```

**Arguments**

|            |                            |
|------------|----------------------------|
| res        | enrichment analysis result |
| geneSymbol | the candidate gene         |

**Value**

subset of enrichment results

**Examples**

```
data("exampleEaRes")
obtainPath(res = exampleEaRes, geneSymbol="ERCC7")
```

---

|               |                      |
|---------------|----------------------|
| queryCpDistLs | <i>queryCpDistLs</i> |
|---------------|----------------------|

---

### Description

produce a plot of `bnlearn::cpdist` by performing `bnlearn::cpdist` on specified node, evidence and level.

### Usage

```
queryCpDistLs(fitted, candidate, evidences, discPalette = "Set2", ...)
```

### Arguments

|                          |  |
|--------------------------|--|
| <code>fitted</code>      | <code>bn.fit</code> object                               |
| <code>candidate</code>   | name of node   |
| <code>evidences</code>   | the evidences  |
| <code>discPalette</code> | palette to be used for plotting if the event is discrete |
| <code>...</code>         | other parameters passed to <code>bnlearn cpdist</code>   |

### Value

list of dataframe containing raw values

### Examples

```
library(bnlearn)
data("exampleEaRes")
data("exampleGeneExp")
net <- bngeneplot(exampleEaRes, exampleGeneExp,
                 pathNum=1, returnNet=TRUE)
fitted <- bn.fit(net$av, net$df)
res <- queryCpDistLs(fitted, candidate="ERCC4",
                   evidences=c("ERCC2<0.1", "ERCC2>0.5", "ERCC2>0.8"), n=500)
```

---

|               |                      |
|---------------|----------------------|
| queryCpDistLw | <i>queryCpDistLw</i> |
|---------------|----------------------|

---

### Description

produce a plot of `bnlearn::cpdist` by performing `bnlearn::cpdist` on specified node, evidence and level.

**Usage**

```
queryCpDistLw(  
  fitted,  
  candidate,  
  evidence,  
  levels,  
  point = FALSE,  
  pointSize = 5,  
  alpha = TRUE,  
  ...  
)
```

**Arguments**

|           |   |
|-----------|---|
| fitted    | bn.fit object   |
| candidate | name of node  |
| evidence  | evidence variable name  |
| levels    | level to be listed  |
| point     | geom_point the weighted mean                                    |
| pointSize | point size for geom_point                                       |
| alpha     | whether to reflect the weights by alpha (TRUE) or color (FALSE) |
| ...       | other parameters passed to bnlearn cpdist                       |

**Value**

list of dataframe containing raw values

**Examples**

```
library(bnlearn)  
data("exampleEaRes")  
data("exampleGeneExp")  
net <- bngeneplot(exampleEaRes, exampleGeneExp,  
  pathNum=1, returnNet=TRUE)  
fitted <- bn.fit(net$av, net$df)  
res <- queryCpDistLw(fitted, candidate="ERCC4", evidence="ERCC2",  
  levels=c(0.1, 0.5, 0.8), n=500)
```

# Index

## \* datasets

exampleEaRes, [17](#)  
exampleGeneExp, [18](#)

## \* internal

CBNplot-package, [2](#)

bngenepplot, [3](#)

bngenepplotCustom, [6](#)

bngenetest, [9](#)

bnpathplot, [10](#)

bnpathplotCustom, [12](#)

bnpathtest, [15](#)

CBNplot (CBNplot-package), [2](#)

CBNplot-package, [2](#)

compareBNs, [17](#)

exampleEaRes, [17](#)

exampleGeneExp, [18](#)

inferMS, [18](#)

loadSign, [19](#)

obtainPath, [19](#)

queryCpDistLs, [20](#)

queryCpDistLw, [20](#)