Package 'theoRy'

January 14, 2024

Title An R package for comparing many causal models

Version 0.1.0

Description This package helps researchers build all possible causal models from variables and efficiently compare them. The package is built on top of the ``dagitty" and ``ggdag" packages, which are already widely used in causal inferences.

License MIT + file LICENSE

Encoding UTF-8

URL https://github.com/hungnguyen167/theoRy

R topics documented:

`add_compatible` *Add test compatible or full model compatible columns to the formula matrix*

Description

`add_compatible` returns a comparison matrix, which is a formula matrix with both the test compatible and full model compatible columns. The user is strongly recommended to pick a reference model which they want to compare against the model universe. If not chosen, reference model is default to the first model in the formula matrix.

Usage

```
add_compatible(formula_matrix, effect = "direct", ref_mod = NULL)
```

Arguments

formula_matrix	the input formula matrix. Created from build_formula_matrix
effect	Effect type for computing the minimum adjustment sets (MAS). See also adjustmentSets
ref_mod	the input reference model. Should avoid using models with correct_test=="no" (incorrectly adjusted) as the reference model.

Details

add_compatible requires both correct_test and formula to be present in the formula matrix. This is the default behavior when using build_formula_matrix to create the formula matrix.

Value

A comparison matrix (data.table format) with two notable columns: test_compatible and full_model_compatible. Consult the paper that goes along with the package for a deeper understanding of what test_compatible and full_model_compatible are.

References

TBA

Examples

```
cmp_matrix <- add_compatible(formula_matrix, effect="direct", ref_mod=1)
```

---

build_causal_node *Build causal matrix or build node-timing matrix*

Description

build_causal_node returns either the causal matrix or the node-timing matrix used in theory comparison

Usage

```
build_causal_node(
  nodes,
  types,
  timing,
  user_mods = NULL,
  include_subsets = FALSE,
  return_node = FALSE
)
```

Arguments

nodes	the input nodes/variable names, a character vector/list. Should be of the same length and order with types and timing.
types	the input types of nodes, a character vector/list. Takes only three values: "otc" (outcome), "ctr" (control), or "test" (test). Should be of the same length and order with nodes and timing.
timing	the input timing of nodes, a character vector/list. Should be of the same length and order with nodes and types.
user_mods	User-defined model(s), optional. This argument allows users to input their theoretical biases. If the model(s) are found in the matrix, they will be pushed to the top in the orders that they are introduced. If the model(s) are not found in the matrix, the user can choose to add it to the matrix or not.
include_subsets	if TRUE, the matrix will include models where certain nodes (besides outcome and exposure) do not exist. Note that model A where X does not cause anything but still exists is a different theoretical claim than model B where X does not exist entirely. The matrix can get significantly larger when this option is allowed. Default to FALSE.
return_node	if TRUE, returns the node-timing matrix instead of the causal matrix.

Details

This function is included in theoRy. Unless computing the causal matrix separately is require, users are encouraged to use theoRy instead.

Value

Either a causal matrix or a node-timing matrix. The causal matrix is the basis for comparing theoretical models. The causal matrix consists of at least 9 columns: from, to, direction, model, component, timing_from, type_from, timing_to, type_to. The direction only takes two values "->" or "<->". "<-" is omitted because X2 <- X1 is similar to X1 -> X2. The column user_mod is introduced when user_mods are provided. In the causal matrix, nodes are called by conventional node names for causal inference (Y, Xtest, X1, X2, etc.) instead of their original variable names. A summary of variable names, timing, and types is provided when the function runs. Users can check this information in the node-timing matrix.

Examples

```
nodes <- c("y","xtest","ctr1","ctr2")
timing <- c(0,-1,-3,-2)
types <- c("otc","test","ctr","ctr")
user_mods <- c("y ~ xtest + ctr2; xtest ~ ctr1 + ctr2", "y ~ xtest + ctr1; xtest ~ ctr1 + ctr2")
causal_matrix <- build_causal_node(nodes=nodes, types=types, timing=timing, user_mods=user_mods, include_subs
```

build_formula_matrix *Build formula matrix*

Description

build_formula_matrix creates a formula matrix (in the lavaan format) from an input causal matrix.

Usage

```
build_formula_matrix(causal_matrix)
```

Arguments

causal_matrix	the input causal matrix. Created from build_causal_matrix

Details

This function is included in theoRy. Unless computing the formula matrix separately is require, users are encouraged to use theoRy instead.

Value

A formula matrix with 5 columns. The formula column is in the lavaan format. See . The MAS column is the minimum adjustment sets to measure the direct effect from Xtest to Y. correct_test (yes or no) is whether the model is correctly adjusted when all X variables are adjusted. This is useful for add_compatible later.

Examples

```
formula_matrix <- build_formula_matrix(causal_matrix)
```

build_set_matrix *Build a set matrix ready for set theory analysis out of the causal matrix*

Description

build_set_matrix creates a set matrix used in theory analysis

Usage

```
build_set_matrix(
  causal_matrix,
  outcome_var = "test_compatible",
  outcome_positive = "compatible",
  cmp_matrix
)
```

Arguments

causal_matrix	the input causal matrix. Created from build_causal_matrix.
outcome_var	the type of compatibility used in comparison. Default to "test_compatible".
outcome_positive	the label for a positive outcome. Default to "compatible".
cmp_matrix	the compatibility matrix. Created from add_compatible.

Value

A formula matrix with 5 columns. The formula column is in the lavaan format. See . The MAS column is the minimum adjustment sets to measure the direct effect from Xtest to Y. correct_test (yes or no) is whether the model is correctly adjusted when all X variables are adjusted. This is useful for add_compatible later.

Examples

```
set_matrix <- build_set_matrix(causal_matrix=causal_matrix, cmp_matrix=cmp_matrix)
```

---

find_add_models *Find or add models to the model universe*

Description

find_add_models allows users to search for a particular model within the model universe. When not found, users can choose to add it to the universe to compare it with existing models.

Usage

```
find_add_models(
  ls_theory = NULL,
  causal_matrix = NULL,
  node_timing = NULL,
  user_mods,
  on_ls = TRUE,
  add_nodes = NULL,
  assert_mod_num = NULL
)
```

Arguments

ls_theory	the input ls_theory object. Created from theoRy.
causal_matrix	the input causal_matrix. Created from build_causal_matrix. Used only when ls_theory=NULL.
node_timing	the input node_timing Created from build_causal_matrix. Used only when ls_theory=NULL.
user_mods	the user's input model(s). Must be in lavaan format. See .
on_ls	whether to use ls_theory or causal matrix. Default to TRUE (use ls_theory).

add_nodes	if the user-defined models have extra nodes/timing/types, should be provided as a list with three named elements: nodes, types, and timing.
assert_mod_num	model numbers to assert on user-defined models. When not provided, added models will be appended to the end of the matrices.

Value

DAG plots of chosen DAG models from the ls_theory.

Examples

```
ls_theory <- find_add_models(ls_theory = ls_theory, on_ls = TRUE,
user_mods = c("y ~ xtest + ctr2; xtest ~ ctr3 + ctr2", # Model 1
"y ~ xtest + ctr2; xtest ~ ctr3 + ctr2", # Model 2
"y ~ xtest + ctr3 + ctr2; xtest ~ ctr3 + ctr2", # Model 3
"y ~ xtest + ctr3; xtest ~ ctr3 + ctr2; ctr4  ~ y + xtest", # Model 4
"y ~ xtest + ctr3; xtest ~ ctr3; ctr2 ~ ctr2", # Model 5
"y ~ xtest + ctr3; xtest ~ ctr2 + ctr1 + ctr3; ctr1 ~ ctr3" # Model 6),
assert_mod_num = c(1,2,3,4,5,6))
```

---

plot_dag *Plot DAG models*

Description

plot_dag creates ggplot2-style plots from a ls_theory object.

Usage

```
plot_dag(ls_theory, choose_plots = "all", save_path = NULL)
```

Arguments

ls_theory	the input ls_theory object. Created from theoRy.
choose_plots	models to plot. Default to "all". However, this option can be resource-intensive if the model universe is too large. It is recommended to choose certain models to compare against one another.
save_path	path to save plots. Default to NULL (not saving)

Details

This requires the ls_theory object, created from theoRy to work. Plot functions are inhereted from the package.

Value

DAG plots of chosen DAG models from the ls_theory.

References

TBA

Examples

```
dag_plots <- plot_dag(ls_theory, choose_plots=c(1,2,3,4,5,6))
for (i in seq_along(dag_plots)){
    print(dag_plots[[i]])
}
```

---

theoRy *A wrapper function to build necessary matrices for theory comparison*

Description

This function returns three main matrices required in theory comparison, namely the node-timing matrix (user inputs), the causal matrix, and the formula matrix. See build_causal_matrix and build_formula_matrix.

Usage

```
theoRy(nodes, types, timing, include_subsets = FALSE, user_mods = NULL)
```

Arguments

nodes	the input nodes/variable names, a character vector/list. Should be of the same length and order with types and timing.
types	the input types of nodes, a character vector/list. Takes only three values: "otc" (outcome), "ctr" (control), or "test" (test). Should be of the same length and order with nodes and timing.
timing	the input timing of nodes, a character vector/list. Should be of the same length and order with nodes and types.
include_subsets	if TRUE, the matrix will include models where certain nodes (besides outcome and exposure) do not exist. Note that model A where X does not cause anything but still exists is a different theoretical claim than model B where X does not exist entirely. The matrix can get significantly larger when this option is allowed. Default to FALSE.
user_mods	User-defined model(s), optional. This argument allows users to input their theoretical biases. If the model(s) are found in the matrix, they will be pushed to the top in the orders that they are introduced. If the model(s) are not found in the matrix, the user can choose to add it to the matrix or not.

Value

a list with three objects: causal_matrix, formula_matrix, and node_timing matrix.

Examples

```
nodes <- c("y","xtest","ctr1","ctr2")
timing <- c(0,-1,-3,-2)
types <- c("otc","test","ctr","ctr")
user_mods <- c("y ~ xtest + ctr2; xtest ~ ctr1 + ctr2", "y ~ xtest + ctr1; xtest ~ ctr1 + ctr2")
ls_theory <- theoRy(nodes=nodes, types=types, timing=timing, include_subsets=TRUE, user_mods=user_mods)
```

Index