Package ‘tpc’

May 8, 2026

Type Package

Title Tiered PC Algorithm

Version 1.0

Maintainer Ronja Foraita <foraita@leibniz-bips.de>

Description Constraint-based causal discovery using the PC algorithm while
accounting for a partial node ordering, for example a partial temporal ordering
when the data were collected in different waves of a cohort study.
Andrews RM, Foraita R, Didelez V, Witte J (2021) <doi:10.48550/arXiv.2108.13395>
provide a guide how to use tpc to analyse cohort data.

Depends pcalg, R (>= 3.5.0)

Imports graph, graphics, methods, parallel, utils

Suggests Rgraphviz, testthat (>= 3.0.0)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

License GPL (>= 3)

URL https://github.com/bips-hb/tpc

BugReports https://github.com/bips-hb/tpc/issues

Config/testthat/edition 3

NeedsCompilation no

Author Janine Witte [aut],
Ronja Foraita [cre, ctb] (ORCID:
<https://orcid.org/0000-0003-2216-6653>),
DFG [fnd]

Repository CRAN

Date/Publication 2023-02-20 11:40:02 UTC

Contents

`dat_cohort` *Simulated Cohort Data*

Description

Simulated data based on 'true_sim' of a European child-and-youth cohort study with three waves (`t0`, `t1` and `t2`). See Andrews et al. (2021) <https://arxiv.org/abs/2108.13395> for more information on how the data were generated.

Usage

```
dat_cohort
```

Format

A data frame with 5000 observations and 34 variables (10 variables were measured at three time points each, denoted as "_t0", "_t1" and "_t2").

sex Sex. Factor variable with levels "male" and "female".

country Country of residence. Factor variable with levels "ITA", "EST", "CYP", "BEL", "SWE", "GER", "HUN" and "ESP".

fto Genotype of one SNP located in the FTO gene. Factor variable with levels "TT", "AT", "AA".

birth_weight Birth weight in grams (numeric).

age_t0 Age in years at survey 't0' (numeric).

age_t1 Age in years at survey 't1' (numeric).

age_t2 Age in years at survey 't2' (numeric).

bmi_t0 Body mass index z-score adjusted for sex and age at survey 't0' (numeric).

bmi_t1 Body mass index z-score adjusted for sex and age at survey 't1' (numeric).

bmi_t2 Body mass index z-score adjusted for sex and age at survey 't2' (numeric).

bodyfat_t0 Per cent body fat measured at survey 't0' (numeric).

bodyfat_t1 Per cent body fat measured at survey 't1' (numeric).

bodyfat_t2 Per cent body fat measured at survey 't2' (numeric).

education_t0 Educational level at survey 't0'. Factor variable with levels "low education", "medium education" and "high education".

education_t1 Educational level at survey 't1'. Factor variable with levels "low education", "medium education" and "high education".

education_t2 Educational level at survey 't2'. Factor variable with levels "low education", "medium education" and "high education".

fiber_t0 Fiber intake in log(mg/kcal) at survey 't0' (numeric).

fiber_t1 Fiber intake in log(mg/kcal) at survey 't1' (numeric).

fiber_t2 Fiber intake in log(mg/kcal) at survey 't2' (numeric).

media_devices_t0 Number of audiovisual media in the child's bedroom at survey 't0' (numeric).

media_devices_t1 Number of audiovisual media in the child's bedroom at survey 't1' (numeric).

media_devices_t2 Number of audiovisual media in the child's bedroom at survey 't2' (numeric).

media_time_t0 Use of audiovisual media in log(h/week+1) at survey 't0' (numeric)

media_time_t1 Use of audiovisual media in log(h/week+1) at survey 't1' (numeric)

media_time_t2 Use of audiovisual media in log(h/week+1) at survey 't2' (numeric)

mvpa_t0 Moderate to vigorous physical activity in sqrt(min/day) at survey 't0' (numeric).

mvpa_t1 Moderate to vigorous physical activity in sqrt(min/day) at survey 't1' (numeric).

mvpa_t2 Moderate to vigorous physical activity in sqrt(min/day) at survey 't2' (numeric).

sugar_t0 Square root of sugar intake score at survey 't0' (numeric).

sugar_t1 Square root of sugar intake score at survey 't1' (numeric).

sugar_t2 Square root of sugar intake score at survey 't2' (numeric).

wellbeing_t0 Box-Cox-transformed well-being score at survey 't0' (numeric).

wellbeing_t1 Box-Cox-transformed well-being score at survey 't1' (numeric).

wellbeing_t2 Box-Cox-transformed well-being score at survey 't2' (numeric).

References

Andrews RM, Foraita R, Witte J (2021). A practical guide to causal discovery with cohort data. <https://doi.org/10.48550/arXiv.2108.13395>

See Also

[tpc::dat_cohort_dis()], [tpc::dat_cohort_mis()]

dat_cohort_dis *Simulated Cohort Data - discretized*

Description

Data from dat_cohort for which all continuous variables have been categorized into three categories.

Usage

```
dat_cohort_dis
```

Format

A data frame with 5000 observations and 34 variables (10 variables were measured at three time points each, denoted as "_t0", "_t1" and "_t2").

sex Sex. Factor variable with levels "male" and "female".

country Country of residence. Factor variable with levels "ITA", "EST", "CYP", "BEL", "SWE", "GER", "HUN" and "ESP".

fto Genotype of one SNP located in the FTO gene. Factor variable with levels "TT", "AT", "AA".

birth_weight Birth weight in grams (numeric).

age_t0 Age in years at survey 't0' (numeric).

age_t1 Age in years at survey 't1' (numeric).

age_t2 Age in years at survey 't2' (numeric).

bmi_t0 Body mass index z-score adjusted for sex and age at survey 't0' (numeric).

bmi_t1 Body mass index z-score adjusted for sex and age at survey 't1' (numeric).

bmi_t2 Body mass index z-score adjusted for sex and age at survey 't2' (numeric).

bodyfat_t0 Per cent body fat measured at survey 't0' (numeric).

bodyfat_t1 Per cent body fat measured at survey 't1' (numeric).

bodyfat_t2 Per cent body fat measured at survey 't2' (numeric).

education_t0 Educational level at survey 't0'. Factor variable with levels "low education", "medium education" and "high education".

education_t1 Educational level at survey 't1'. Factor variable with levels "low education", "medium education" and "high education".

education_t2 Educational level at survey 't2'. Factor variable with levels "low education", "medium education" and "high education".

fiber_t0 Fiber intake in log(mg/kcal) at survey 't0' (numeric).

fiber_t1 Fiber intake in log(mg/kcal) at survey 't1' (numeric).

fiber_t2 Fiber intake in log(mg/kcal) at survey 't2' (numeric).

media_devices_t0 Number of audiovisual media in the child's bedroom at survey 't0' (numeric).

media_devices_t1 Number of audiovisual media in the child's bedroom at survey 't1' (numeric).

media_devices_t2 Number of audiovisual media in the child's bedroom at survey 't2' (numeric).

media_time_t0 Use of audiovisual media in log(h/week+1) at survey 't0' (numeric)

media_time_t1 Use of audiovisual media in log(h/week+1) at survey 't1' (numeric)

media_time_t2 Use of audiovisual media in log(h/week+1) at survey 't2' (numeric)

mvpa_t0 Moderate to vigorous physical activity in sqrt(min/day) at survey 't0' (numeric).

mvpa_t1 Moderate to vigorous physical activity in sqrt(min/day) at survey 't1' (numeric).

mvpa_t2 Moderate to vigorous physical activity in sqrt(min/day) at survey 't2' (numeric).

sugar_t0 Square root of sugar intake score at survey 't0' (numeric).

sugar_t1 Square root of sugar intake score at survey 't1' (numeric).

sugar_t2 Square root of sugar intake score at survey 't2' (numeric).

wellbeing_t0 Box-Cox-transformed well-being score at survey 't0' (numeric).

wellbeing_t1 Box-Cox-transformed well-being score at survey 't1' (numeric).

wellbeing_t2 Box-Cox-transformed well-being score at survey 't2' (numeric).

References

Andrews RM, Foraita R, Witte J (2021). A practical guide to causal discovery with cohort data. <https://doi.org/10.48550/arXiv.2108.13395>

See Also

[tpc::dat_cohort()], [tpc::dat_cohort_mis()]

`dat_cohort_mis` *Simulated Cohort Data - with missing values*

Description

Data from `dat_cohort` with missing values.

Usage

```
dat_cohort_mis
```

Format

A data frame with 5000 observations and 34 variables (10 variables were measured at three time points each, denoted as "_t0", "_t1" and "_t2").

sex Sex. Factor variable with levels "male" and "female".

country Country of residence. Factor variable with levels "ITA", "EST", "CYP", "BEL", "SWE", "GER", "HUN" and "ESP".

fto Genotype of one SNP located in the FTO gene. Ordinal variable with levels "TT", "AT", "AA".

birth_weight Birth weight in grams (numeric).

age_t0 Age in years at survey ’t0’ (numeric).

age_t1 Age in years at survey ’t1’ (numeric).

age_t2 Age in years at survey ’t2’ (numeric).

bmi_t0 Body mass index z-score adjusted for sex and age at survey ’t0’ (numeric).

bmi_t1 Body mass index z-score adjusted for sex and age at survey ’t1’ (numeric).

bmi_t2 Body mass index z-score adjusted for sex and age at survey ’t2’ (numeric).

bodyfat_t0 Per cent body fat measured at survey ’t0’ (numeric).

bodyfat_t1 Per cent body fat measured at survey ’t1’ (numeric).

bodyfat_t2 Per cent body fat measured at survey ’t2’ (numeric).

education_t0 Educational level at survey ’t0’. Ordinal variable with levels "low education", "medium education" and "high education".

education_t1 Educational level at survey ’t1’. Ordinal variable with levels "low education", "medium education" and "high education".

education_t2 Educational level at survey ’t2’. Ordinal variable with levels "low education", "medium education" and "high education".

fiber_t0 Fiber intake in log(mg/kcal) at survey ’t0’ (numeric).

fiber_t1 Fiber intake in log(mg/kcal) at survey ’t1’ (numeric).

fiber_t2 Fiber intake in log(mg/kcal) at survey ’t2’ (numeric).

media_devices_t0 Number of audiovisual media in the child’s bedroom at survey ’t0’ (numeric).

media_devices_t1 Number of audiovisual media in the child’s bedroom at survey ’t1’ (numeric).

media_devices_t2 Number of audiovisual media in the child’s bedroom at survey ’t2’ (numeric).

media_time_t0 Use of audiovisual media in log(h/week+1) at survey ’t0’ (numeric)

media_time_t1 Use of audiovisual media in log(h/week+1) at survey ’t1’ (numeric)

media_time_t2 Use of audiovisual media in log(h/week+1) at survey ’t2’ (numeric)

mvpa_t0 Moderate to vigorous physical activity in sqrt(min/day) at survey ’t0’ (numeric).

mvpa_t1 Moderate to vigorous physical activity in sqrt(min/day) at survey ’t1’ (numeric).

mvpa_t2 Moderate to vigorous physical activity in sqrt(min/day) at survey ’t2’ (numeric).

sugar_t0 Square root of sugar intake score at survey ’t0’ (numeric).

sugar_t1 Square root of sugar intake score at survey ’t1’ (numeric).

sugar_t2 Square root of sugar intake score at survey ’t2’ (numeric).

wellbeing_t0 Box-Cox-transformed well-being score at survey ’t0’ (numeric).

wellbeing_t1 Box-Cox-transformed well-being score at survey ’t1’ (numeric).

wellbeing_t2 Box-Cox-transformed well-being score at survey ’t2’ (numeric).

References

Andrews RM, Foraita R, Witte J (2021). A practical guide to causal discovery with cohort data. <https://doi.org/10.48550/arXiv.2108.13395>

See Also

[tpc::dat_cohort()], [tpc::dat_cohort_dis()]

`dat_sim` — *Simulated Data with a Partial Ordering*

Description

A simple graph and corresponding dataset used in the examples illustrating `tpc`.

Usage

```
dat_sim
```

Format

A data frame with 1000 observations and 9 numerical variables simulated by drawing from a multivariate distribution according to the DAG `true_sim`.

A1 numeric

B1 numeric

C1 numeric

A2 numeric

B2 numeric

C2 numeric

A3 numeric

B3 numeric

C3 numeric

`MeekRules` — *Last Step of tPC Algorithm: Apply Meek's rules*

Description

This is a modified version of `pcalg::udag2pdagRelaxed`. It applies Meek's rules to the partially oriented graph obtained after orienting edges between time points / tiers.

Usage

```
MeekRules(
  gInput,
  verbose = FALSE,
  unfVect = NULL,
  solve.confl = FALSE,
  rules = rep(TRUE, 4)
)
```

Arguments

`gInput` 'pcAlgo'-object containing skeleton and conditional indepedence information.

`verbose` FALSE: No output; TRUE: Details

`unfVect` Vector containing numbers that encode ambiguous triples (as returned by [tpc_cons_intern()]. This is needed in the conservative and majority rule PC algorithms.

`solve.confl` If `TRUE`, the orientation rules work with lists for candidate sets and allow bi-directed edges to resolve conflicting edge orientations. Note that therefore the resulting object is order-independent but might not be a PDAG because bi-directed edges can be present.

`rules` A vector of length 4 containing `TRUE` or `FALSE` for each rule. `TRUE` in position i means that rule i (Ri) will be applied. By default, all rules are used.

Details

If `unfVect` = `NULL` (no ambiguous triples), the four orientation rules are applied to each eligible structure until no more edges can be oriented. Otherwise, unfVect contains the numbers of all ambiguous triples in the graph as determined by [tpc_cons_intern()]. Then the orientation rules take this information into account. For example, if `a -> b - c` and `<a,b,c>` is an unambigous triple and a non-v-structure, then rule 1 implies `b -> c`. On the other hand, if `a -> b - c` but `<a,b,c>` is an ambiguous triple, then the edge `b - c` is not oriented.

If `solve.confl` = `FALSE`, earlier edge orientations are overwritten by later ones.

If `solv.confl` = `TRUE`, both the v-structures and the orientation rules work with lists for the candidate edges and allow bi-directed edges if there are conflicting orientations. For example, two v-structures `a -> b <- c` and `b -> c <- d` then yield `a -> b <-> c <- d`. This option can be used to get an order-independent version of the PC algorithm (see Colombo and Maathuis (2014)).

We denote bi-directed edges, for example between two variables i and j, in the adjacency matrix M of the graph as `M[i,j]=2` and `M[j,i]=2`. Such edges should be interpreted as indications of conflicts in the algorithm, for example due to errors in the conditional independence tests or violations of the faithfulness assumption.

Value

An object of class `pcAlgo-class`.

Author(s)

Original code by Markus Kalisch, modifications by Janine Witte.

References

C. Meek (1995). Causal inference and causal explanation with background knowledge. In: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI-95), pp. 403-411. Morgan Kaufmann Publishers.

D. Colombo and M.H. Maathuis (2014). Order-independent constraint-based causal structure learning. Journal of Machine Learning Research 15:3741-3782.

Examples

```
data(dat_sim)
sk.fit <- skeleton(suffStat = list(C = cor(dat_sim), n = nrow(dat_sim)),
            indepTest = gaussCItest, labels = names(dat_sim), alpha = 0.05)
MeekRules(sk.fit)
```

---

tpc *PC Algorithm Accounting for a Partial Node Ordering*

Description

Like [pcalg::pc()], but takes into account a user-specified partial ordering of the nodes/variables. This has two effects: 1) The conditional independence between `x` and `y` given `S` is ot tested if any variable in `S` lies in the future of both `x` and `y`; 2) edges cannot be oriented from a higher-order to a lower-order node. In addition, the user may specify individual forbidden edges and context variables.

Usage

```
tpc(
  suffStat,
  indepTest,
  alpha,
  labels,
  p,
  skel.method = c("stable", "stable.parallel"),
  forbEdges = NULL,
  m.max = Inf,
  conservative = FALSE,
  maj.rule = TRUE,
  tiers = NULL,
  context.all = NULL,
  context.tier = NULL,
  verbose = FALSE,
  numCores = NULL,
  cl.type = "PSOCK",
  clusterexport = NULL
)
```

Arguments

`suffStat` A [base::list()] of sufficient statistics, containing all necessary elements for the conditional independence decisions in the function [indepTest()].

`indepTest` A function for testing conditional independence. It is internally called as `indepTest(x,y,S,suffStat)`, and tests conditional independence of `x` and `y` given `S`. Here, `x` and `y` are variables, and `S` is a (possibly empty) vector of variables (all variables are denoted

	by their (integer) column positions in the adjacency matrix). `suffStat` is a list, see the argument above. The return value of `indepTest` is the p-value of the test for conditional independence.
`alpha`	significance level (number in *(0,1)* for the individual conditional independence tests.
`labels`	(optional) character vector of variable (or "node") names. Typically preferred to specifying `p`.
`p`	(optional) number of variables (or nodes). May be specified if `labels` are not, in which case `labels` is set to `1:p`.
`skel.method`	Character string specifying method; the default, "stable" provides an order-independent skeleton, see [tpc::tskeleton()].
`forbEdges`	A logical matrix of dimension p*p. If `[i,j]` is TRUE, then the directed edge i->j is forbidden. If both `[i,j]` and `[j,i]` are TRUE, then any type of edge between i and j is forbidden.
`m.max`	Maximal size of the conditioning sets that are considered in the conditional independence tests.
`conservative`	Logical indicating if conservative PC should be used. Defaults to FALSE. See [pcalg::pc()] for details.
`maj.rule`	Logical indicating if the majority rule should be used. Defaults to TRUE. See [pcalg::pc()] for details.
`tiers`	Numeric vector specifying the tier / time point for each variable. Must be of length 'p', if specified, or have the same length as 'labels', if specified. A smaller number corresponds to an earlier tier / time point.
`context.all`	Numeric or character vector. Specifies the positions or names of global context variables. Global context variables have no incoming edges, i.e. no parents, and are themselves parents of all non-context variables in the graph.
`context.tier`	Numeric or character vector. Specifies the positions or names of tier-specific context variables. Tier-specific context variables have no incoming edges, i.e. no parents, and are themselves parents of all non-context variables in the same tier.
`verbose`	if `TRUE`, detailed output is provided.
`numCores`	The numbers of CPU cores to be used.
`cl.type`	The cluster type. Default value is `"PSOCK"`. For High-performance clusters use `"MPI"`. See also `parallel::makeCluster`.
`clusterexport`	Character vector. Lists functions to be exported to nodes if numCores > 1.

Details

See `pcalg::pc` for further information on the PC algorithm. The PC algorithm is named after its developers Peter Spirtes and Clark Glymour (Spirtes et al., 2000).

Specifying a tier for each variable using the `tier` argument has the following effects: 1) In the skeleton phase and v-structure learing phases, conditional independence testing is restricted such that if x is in tier t(x) and y is in t(y), only those variables are allowed in the conditioning set whose tier is not larger than t(x). 2) Following the v-structure phase, all edges that were found between two tiers are directed into the direction of the higher-order tier. If context variables are specified using `context.all` and/or `context.tier`, the corresponding orientations are added in this step.

Value

An object of class "pcAlgo" (see [pcalg::pcalgo] containing an estimate of the equivalence class of the underlying DAG.

Author(s)

Original code by Markus Kalisch, Martin Maechler, and Diego Colombo. Modifications by Janine Witte (Kalisch et al., 2012).

References

M. Kalisch, M. Maechler, D. Colombo, M.H. Maathuis and P. Buehlmann (2012). Causal Inference Using Graphical Models with the R Package pcalg. Journal of Statistical Software 47(11): 1–26.

P. Spirtes, C. Glymour and R. Scheines (2000). Causation, Prediction, and Search, 2nd edition. The MIT Press. https://philarchive.org/archive/SPICPA-2.

Examples

```
# load simulated cohort data
data(dat_sim)
n <- nrow(dat_sim)
lab <- colnames(dat_sim)

# estimate skeleton without taking background information into account
tpc.fit <- tpc(suffStat = list(C = cor(dat_sim), n = n),
               indepTest = gaussCItest, alpha = 0.01, labels = lab)
pc.fit <- pcalg::pc(suffStat = list(C = cor(dat_sim), n = n),
                    indepTest = gaussCItest, alpha = 0.01, labels = lab,
                    maj.rule = TRUE, solve.conf = TRUE)
identical(pc.fit@graph, tpc.fit@graph) # TRUE
# estimate skeleton with temporal ordering as background information
tiers <- rep(c(1,2,3), times=c(3,3,3))
tpc.fit2 <- tpc(suffStat = list(C = cor(dat_sim), n = n),
                indepTest = gaussCItest, alpha = 0.01, labels = lab, tiers = tiers)

tpc.fit3 <- tpc(suffStat = list(C = cor(dat_sim), n = n),
                indepTest = gaussCItest, alpha = 0.01, labels = lab, tiers = tiers,
                skel.method = "stable.parallel",
                numCores = 2, clusterexport = c("cor", "ecdf"))

if(requireNamespace("Rgraphviz", quietly = TRUE)){
 data("true_sim")
 oldpar <- par(mfrow = c(1,3))
 plot(true_sim, main = "True DAG")
 plot(tpc.fit, main = "PC estimate")
 plot(tpc.fit2, main = "tPC estimate")
 par(oldpar)
 }

 # require that there is no edge between A1 and A1, and that any edge between A2 and B2
 # or A2 and C2 is directed away from A2
```

```
forb <- matrix(FALSE, nrow=9, ncol=9)
rownames(forb) <- colnames(forb) <- lab
forb["A1","A3"] <- forb["A3","A1"] <- TRUE
forb["B2","A2"] <- TRUE
forb["C2","A2"] <- TRUE

tpc.fit3 <- tpc(suffStat = list(C = cor(dat_sim), n = n),
                indepTest = gaussCItest, alpha = 0.01,labels = lab,
                forbEdges = forb, tiers = tiers)

if (requireNamespace("Rgraphviz", quietly = TRUE)) {
# compare estimated CPDAGs
  data("true_sim")
  oldpar <- par(mfrow = c(1,2))
  plot(tpc.fit2, main = "old tPC estimate")
  plot(tpc.fit3, main = "new tPC estimate")
  par(oldpar)
}
# force edge from A1 to all other nodes measured at time 1
# into the graph (note that the edge from A1 to A2 is then
# forbidden)
tpc.fit4 <- tpc(suffStat = list(C = cor(dat_sim), n = n),
                indepTest = gaussCItest, alpha = 0.01, labels = lab,
                tiers = tiers, context.tier = "A1")

if (requireNamespace("Rgraphviz", quietly = TRUE)) {
# compare estimated CPDAGs
 data("true_sim")
 plot(tpc.fit4, main = "alternative tPC estimate")
}

# force edge from A1 to all other nodes into the graph
tpc.fit5 <- tpc(suffStat = list(C = cor(dat_sim), n = n),
                indepTest = gaussCItest, alpha = 0.01, labels = lab,
                tiers = tiers, context.all = "A1")

if (requireNamespace("Rgraphviz", quietly = TRUE)) {
# compare estimated CPDAGs
data("true_sim")
plot(tpc.fit5, main = "alternative tPC estimate")
}
```

`tpc.cons.intern` *Utility for Conservative and Majority Rule in tpc*

Description

Like pcalg::pc.cons.intern, but takes into account the user-specified partial node/variable ordering.

Usage

```
tpc.cons.intern(
  sk,
  suffStat,
  indepTest,
  alpha,
  version.unf = c(NA, NA),
  maj.rule = FALSE,
  forbEdges = NULL,
  tiers = NULL,
  context.all = NULL,
  context.tier = NULL,
  verbose = FALSE
)
```

Arguments

`sk`	A skeleton object as returned from `pcalg::skeleton`.
`suffStat`	Sufficient statistic: List containing all relevant elements for the conditional independence decisions.
`indepTest`	Pre-defined `function` for testing conditional independence. The function is internally called as `indepTest(x,y,S,suffStat)`, and tests conditional independence of `x` and `y` given `S`. Here, `x` and `y` are variables, and `S` is a (possibly empty) vector of variables (all variables are denoted by their (integer) column positions in the adjacency matrix). The return value of `indepTest` is the p-value of the test for conditional independence.
`alpha`	Significance level for the individual conditional independence tests.
`version.unf`	Vector of length two. If `version.unf[2]==1`, the inititial separating set found by the PC/FCI algorithm is added to the set of separating sets; if `version.unf[2]==2`, it is not added. In the latter case, if the set of separating sets is empty, the triple is marked as unambiguous if `version.unf[1]==1`, and as ambiguous if `version.unf[1]==2`.
`maj.rule`	Logical indicating if the triples are checked for ambiguity using the majority rule idea, which is less strict than the standard conservative method.
`forbEdges`	A logical matrix of dimension `p*p`. If `[i,j]` is TRUE, then the directed edge `i -> j` is forbidden. If both `[i,j]` and `[j,i]` are TRUE, then any type of edge between `i` and `j` is forbidden.
`tiers`	Numeric vector specifying the tier / time point for each variable. A smaller number corresponds to an earlier tier / time point.
`context.all`	Numeric or character vector. Specifies the positions or names of global context variables. Global context variables have no incoming edges, i.e. no parents, and are themselves parents of all non-context variables in the graph.
`context.tier`	Numeric or character vector. Specifies the positions or names of tier-specific context variables. Tier-specific context variables have no incoming edges, i.e. no parents, and are themselves parents of all non-context variables in the same tier.

`verbose` Logical asking for detailed output.

Details

See `pcalg::pc.cons.intern` for further information on the majority and conservative approaches to learning v-structures.

Specifying a tier for each variable using the `tier` argument has the following effects:

1) Only those triples `x-y-z` are considered as potential v-structures that satisfy `t(y)=max(t(x),t(z))`. This allows for three constellations: either `y` is in the same tier as `x` and both are later than `z`, or `y` is in the same tier as `z` and both are later than `x`, or all three are in the same tier. Triples where `y` is earlier than one or both of `x` and `z` need not be considered, as `y` being a collider would be against the partial ordering. Triples where `y` is later than both `x` and `z` will be oriented later in the pc algorithm and are left out here to minimize the number of conditional independence tests.

2) Conditional independence testing is restricted such that if `x` is in tier `t(x)` and `y` is in `t(y)`, only those variables are allowed in the conditioning set whose tier is not larger than `t(x)`.

Context variables specified via `context.all` or `context.tier` are not considered as candidate colliders or candidate parents of colliders.

Value

unfTripl numeric vector of triples coded as numbers (via `pcalg::triple2numb`) that were marked as ambiguous.

sk The updated skeleton-object (separating sets might have been updated).

Author(s)

Original code by Markus Kalisch and Diego Colombo. Modifications by Janine Witte.

`true_cohort` *Cohort Data Structure*

Description

A DAG from which the data 'data_cohort' was simulated from. See Andrews et al. (2021) <https://arxiv.org/abs/2108.13395> for more information on how the data were generated.

Usage

```
true_cohort
```

Format

A DAG (graphNEL object) with 34 nodes and 128 edges.

References

Andrews RM, Foraita R, Witte J (2021). A practical guide to causal discovery with cohort data. <https://doi.org/10.48550/arXiv.2108.13395>

See Also

See [graph::graphNEL()] for the class 'graphNEL'.

---

true_sim — *A DAG with a Partial Ordering*

Description

An example DAG from which the data 'data_sim' was simulated from.

Usage

```
true_sim
```

Format

A DAG (graphNEL object) with 9 nodes and 7 edges.

See Also

See [graph::graphNEL()] for the class 'graphNEL'.

---

tskeleton — *Estimate the Skeleton of a DAG while Accounting for a Partial Ordering*

Description

Like `pcalg::skeleton`, but takes a user-specified partial node ordering into account. The conditional independence between `x` and `y` given `S` is not tested if any variable in `S` lies in the future of both `x` and `y`.

Usage

```
tskeleton(
  suffStat,
  indepTest,
  alpha,
  labels,
  p,
  method = c("stable", "original"),
  m.max = Inf,
  fixedGaps = NULL,
  fixedEdges = NULL,
  NAdelete = TRUE,
  tiers = NULL,
  verbose = FALSE
)
```

Arguments

- `suffStat` — A list of sufficient statistics, containing all necessary elements for the conditional independence decisions in the function `indepTest`.
- `indepTest` — Predefined `function` for testing conditional independence. It is internally called as `indepTest(x,y,S,suffStat)`, and tests conditional independence of `x` and `y` given `S`. Here, `x` and `y` are variables, and `S` is a (possibly empty) vector of variables (all variables are denoted by their (integer) column positions in the adjacency matrix). `suffStat` is a list, see the argument above. The return value of `indepTest` is the p-value of the test for conditional independence.
- `alpha` — Significance level (number in *(0,1)* for the individual conditional independence tests.
- `labels` — (optional) character vector of variable (or "node") names. Typically preferred to specifying `p`.
- `p` — (optional) number of variables (or nodes). May be specified if `labels` are not, in which case `labels` is set to `1:p`.
- `method` — Character string specifying method; the default, "stable" provides an *order-independent* skeleton, see 'Details' below.
- `m.max` — Maximal size of the conditioning sets that are considered in the conditional independence tests.
- `fixedGaps` — logical *symmetric* matrix of dimension `p*p`. If entry `[i,j]` is true, the edge *i-j* is removed before starting the algorithm. Therefore, this edge is guaranteed to be *absent* in the resulting graph.
- `fixedEdges` — a logical *symmetric* matrix of dimension `p*p`. If entry `[i,j]` is true, the edge *i-j* is never considered for removal. Therefore, this edge is guaranteed to be *present* in the resulting graph.
- `NAdelete` — logical needed for the case `indepTest(*)` returns `NA`. If it is true, the corresponding edge is deleted, otherwise not.

`tiers` Numeric vector specifying the tier / time point for each variable. Must be of length 'p', if specified, or have the same length as 'labels', if specified. A smaller number corresponds to an earlier tier / time point. Conditional independence testing is restricted such that if `x` is in tier `t(x)` and `y` is in `t(y)`, only those variables are allowed in the conditioning set whose tier is not larger than `t(x)`.

`verbose` if `TRUE`, detailed output is provided.

Details

See `pcalg::skeleton` for further information on the skeleton algorithm.

Value

An object of class "pcAlgo" (see `pcalg::pcAlgo`) containing an estimate of the skeleton of the underlying DAG, the conditioning sets (sepset) that led to edge removals and several other parameters.

Author(s)

Original code by Markus Kalisch, Martin Maechler, Alain Hauser and Diego Colombo. Modifications by Janine Witte.

Examples

```
# load simulated cohort data
data("dat_sim")
n <- nrow(dat_sim)
lab <- colnames(dat_sim)
# estimate skeleton without taking background information into account
tskel.fit <- tskeleton(suffStat = list(C = cor(dat_sim), n = n),
                       indepTest = gaussCItest, alpha = 0.01, labels = lab)
skel.fit <- pcalg::skeleton(suffStat = list(C = cor(dat_sim), n = n),
                            indepTest = gaussCItest, alpha = 0.01, labels = lab)
                            identical(skel.fit@graph, tskel.fit@graph) # TRUE

# estimate skeleton with temporal ordering as background information
tiers <- rep(c(1,2,3), times=c(3,3,3))
tskel.fit2 <- tskeleton(suffStat = list(C = cor(dat_sim), n = n),
                      indepTest = gaussCItest, alpha = 0.01, labels = lab, tiers = tiers)

# in this case, the skeletons estimated with and without
# background knowledge are identical, but fewer conditional
# independence tests were performed when background
# knowledge was taken into account
identical(tskel.fit@graph, tskel.fit2@graph) # TRUE
tskel.fit@n.edgetests
tskel.fit2@n.edgetests
```

Index