Package 'ciflyr'

July 3, 2025

Title Reachability-Based Primitives for Graphical Causal Inference

Version 0.1.1

Description Provides a framework for specifying and running flexible linear-time reachability-based algorithms for graphical causal inference. Rule tables are used to encode and customize the reachability algorithm to typical causal and probabilistic reasoning tasks such as finding d-connected nodes or more advanced applications. For more information, see Wienöbst, Weichwald and Henckel (2025) <doi:10.48550/arXiv.2506.15758>.

License MIT + file LICENSE

URL https://cifly.pages.dev/, https://github.com/mwien/CIfly

BugReports https://github.com/mwien/CIfly/issues

SystemRequirements Cargo (Rust's package manager), rustc

Depends R (>= 4.2)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/rextendr/version 0.4.0.9000

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation yes

Author Marcel Wienöbst [aut, cre, cph],
Sebastian Weichwald [aut],
Leonard Henckel [aut],
Authors of the dependency Rust crates [ctb] (see inst/AUTHORS file for details)

Maintainer Marcel Wienöbst <marcel.wienoebst@gmx.de>

Repository CRAN

Date/Publication 2025-07-03 13:20:02 UTC

Contents

`parseGraph` *Obtain an internal representation of a CIfly graph.*

Description

Obtain an internal representation of a CIfly graph. Advanced usage only, mostly recommended for improving performance if the same graph is used multiple times. The parsed graph object can be passed to all methods with a graph argument. It is compatible with all ruletables that have the same `EDGES ...` line as the ruletable passed as argument.

Usage

```
parseGraph(graph, ruletable, tableAsString = FALSE)
```

Arguments

`graph`	A list mapping edge types to edge lists.
`ruletable`	Path to a ruletable file.
`tableAsString`	Optional argument to enable passing the ruletable as multi-line string. Default value is FALSE.

Value

Internal CIfly graph representation.

Examples

```
dsepTable <- "
    EDGES --> <--
    SETS X, Z
    START <-- AT X
    OUTPUT ...
    --> | <-- | current in Z
    ... | ... | current not in Z
"
edgelist <- list("-->" = rbind(c(1, 2), c(3, 2), c(2, 4)))

g <- parseGraph(edgelist, dsepTable, tableAsString=TRUE)
sets <- list("X" = c(1), "Z" = c(4))
reach(edgelist, sets, dsepTable, tableAsString=TRUE)
```

parseRuletable *Obtain an internal representation of a CIfly ruletable.*

Description

Obtain an internal representation of a CIfly ruletable. Advanced usage only, mostly recommended for improving performance if the same ruletable is used multiple times. The parsed ruletable object can be passed to all methods with a ruletable argument.

Usage

```
parseRuletable(ruletable, tableAsString = FALSE)
```

Arguments

ruletable	Path to a ruletable file.
tableAsString	Optional argument to enable passing the ruletable as multi-line string. Default value is FALSE.

Value

Internal CIfly ruletable representation.

Examples

```
dsepTable <- "
    EDGES --> <--
    SETS X, Z
    START <-- AT X
    OUTPUT ...
    --> | <-- | current in Z
    ... | ... | current not in Z
"

rt <- parseRuletable(dsepTable, tableAsString=TRUE)
edgelist <- list("-->" = rbind(c(1, 2), c(3, 2), c(2, 4)))
sets <- list("X" = c(1), "Z" = c(4))
reach(edgelist, sets, rt)
```

parseSets *Obtain an internal representation of CIfly sets.*

Description

Obtain an internal representation of CIfly sets. Advanced usage only, mostly recommended for improving performance if the same sets are used multiple times. The parsed sets object can be passed to all methods with a sets argument. It is compatible with all ruletables that have the same `SETS ...` line as the ruletable passed as argument.

Usage

```
parseSets(sets, ruletable, tableAsString = FALSE)
```

Arguments

`sets`	A list mapping set names to a list of elements.
`ruletable`	Path to a ruletable file.
`tableAsString`	Optional argument to enable passing the ruletable as multi-line string. Default value is FALSE.

Value

Internal CIfly sets representation.

Examples

```
dsepTable <- "
    EDGES --> <--
    SETS X, Z
    START <-- AT X
    OUTPUT ...
    --> | <-- | current in Z
    ... | ... | current not in Z
"
sets <- list("X" = c(1), "Z" = c(4))

s <- parseSets(sets, dsepTable, tableAsString=TRUE)
edgelist <- list("-->" = rbind(c(1, 2), c(3, 2), c(2, 4)))
reach(edgelist, s, dsepTable, tableAsString=TRUE)
```

reach *Perform the CIfly algorithm specified in the passed ruletable.*

Description

For the given graph and sets, a CIfly reachability algorithm is run according to the ruletable specified in the ruletable argument. The algorithm returns all reachable nodes. It is guaranteed to run in linear-time.

Usage

```
reach(graph, sets, ruletable, tableAsString = FALSE, verbose = FALSE)
```

Arguments

graph	A list mapping edge types to edge lists stored in matrix format.
sets	A list mapping set names to a list of elements.
ruletable	Path to a ruletable file.
tableAsString	Optional argument to enable passing the ruletable as multi-line string. Default value is FALSE.
verbose	Optional argument to enable logging. Default value is FALSE.

Value

A vector of all reachable nodes.

Examples

```
dsepTable <- "
    EDGES --> <--
    SETS X, Z
    START <-- AT X
    OUTPUT ...
    --> | <-- | current in Z
    ... | ... | current not in Z
"

edgelist <- list("-->" = rbind(c(1, 2), c(3, 2), c(2, 4)))
sets <- list("X" = c(1), "Z" = c(4))
reach(edgelist, sets, dsepTable, tableAsString=TRUE)
```

Index