

Package ‘Trajectories’

November 13, 2021

Type Package

Title Tool to detect and visualize temporal event trajectories in OMOP common data model

Version 1.0.0

Date 2021-11-12

Description Takes events from OMOP CDM database, detects significant directional event pairs and trajectories.

LazyData true

Maintainer Kadri Künnapuu <kadri.kunnapuu@stacc.ee>

License GPL-3

Depends R (>= 3.2.2),
dplyr (>= 1.0.7)

Imports DatabaseConnector (>= 2.2.0),
SqlRender (>= 1.5.2),
stringi (>= 1.4.5),
openxlsx (>= 4.2.4),
jsonlite (>= 1.7.1),
ggplot2 (>= 3.3.3),
igraph (>= 1.2.5),
lubridate (>= 1.7.10),
ParallelLogger (>= 2.0.1),
tidyselect (>= 1.1.1),
epitools (>= 0.5-10.1),
optmatch (>= 0.9-16)

Encoding UTF-8

Suggests testthat (>= 2.1.0)

RoxygenNote 7.1.2

R topics documented:

addBirthsChecker	2
alignActualTrajectoriesToGraph	3
alignActualTrajectoriesToGraphFull	4
createAlignmentTableNew	4
createAndFillCohortTable	5
createCohortTable	5
createEventPairsTable	6

createEventPairsTableBrunak	6
createFilteredFullgraphs	7
createTrajectoriesGraph	7
createTrajectoryAnalysisArgs	8
createTrajectoryLocalArgs	10
createValidationSet	11
createValidationSetup	12
dbCleanup	12
detectSubgraphs	13
discover	13
estimatedTimeRemaining	14
fillCohortTable	15
filterByLift	15
filterIgraphRemoveLowEffectLinksAndOrphanNodes	16
filterTrajectoriesGraphCrossingEvent	16
getCohortSize	17
GetOutputFolder	17
InitLogger	18
insertTable	18
is.TrajectoryAnalysisArgs	19
is.TrajectoryLocalArgs	19
IsValidationMode	19
loadRenderTranslateSql	20
plotTrajectoriesGraph	20
PlotTrajectoriesGraphForEvent	21
PlotTrajectoriesGraphForEvents	22
removeOrphanNodes	23
runDiscoveryAnalysis	23
runValidationAnalysis	24
setRole	24
TrajectoryAnalysisArgsFromInputFolder	25
TrajectoryAnalysisArgsFromJson	25
TrajectoryAnalysisArgsToJson	26
validate	26
visualize_data_pairs	27

Index 28

addBirthsChecker	<i>Checks whether person.birth_datetime is recorded in the data if addBirths is set to TRUE</i>
------------------	---

Description

Checks whether person.birth_datetime is recorded in the data if addBirths is set to TRUE

Usage

```
addBirthsChecker(connection, trajectoryAnalysisArgs, trajectoryLocalArgs)
```

Arguments

connection	Database connection object created by createConnectionDetails() method in DatabaseConnector package
trajectoryAnalysisArgs	TrajectoryAnalysisArgs object that must be created by createTrajectoryAnalysisArgs() method
trajectoryLocalArgs	TrajectoryLocalArgs object that must be created by createTrajectoryLocalArgs() method

alignActualTrajectoriesToGraph

Adds numcohortCustom value to graph edges - actual number of people (out of all people who have eventId) on that edge

Description

Adds numcohortCustom value to graph edges - actual number of people (out of all people who have eventId) on that edge

Usage

```
alignActualTrajectoriesToGraph(
  connection,
  trajectoryAnalysisArgs,
  trajectoryLocalArgs,
  g,
  eventId,
  limit = 1000,
  filename = file.path(getwd(), "trajectories.csv"),
  filename_interpretation = file.path(getwd(), "trajectories_interpretation.txt")
)
```

Arguments

connection	Database connection object created by createConnectionDetails() method in DatabaseConnector package
trajectoryAnalysisArgs	TrajectoryAnalysisArgs object that must be created by createTrajectoryAnalysisArgs() method
trajectoryLocalArgs	TrajectoryLocalArgs object that must be created by createTrajectoryLocalArgs() method
g	An igraph object that is created by specific graph functions in this package
eventId	Event name (concept ID) through which the trajectories are analyzed
limit	Max number of trajectories to align (to limit the analysis). Set to NA if no limit.
filename	Full path to output file for trajectory counts. Set to NA to skip this.
filename_interpretation	Full path to output file for textual interpretation of the output graph. Set to NA to skip this.

alignActualTrajectoriesToGraphFull

Adds numcohortCustom value to graph nodes and edges - actual number of people on that edge/node

Description

Adds numcohortCustom value to graph nodes and edges - actual number of people on that edge/node

Usage

```
alignActualTrajectoriesToGraphFull(
  connection,
  trajectoryAnalysisArgs,
  trajectoryLocalArgs,
  g
)
```

Arguments

connection	Database connection object created by createConnectionDetails() method in DatabaseConnector package
trajectoryAnalysisArgs	TrajectoryAnalysisArgs object that must be created by createTrajectoryAnalysisArgs() method
trajectoryLocalArgs	TrajectoryLocalArgs object that must be created by createTrajectoryLocalArgs() method
g	An igraph object that is created by specific graph functions in this package

createAlignmentTableNew

Creates and fills alignment tables to the database for a given graph. Particularly, creates table "graph_events" for events and table "graph_event_pairs" for the pairs

Description

Creates and fills alignment tables to the database for a given graph. Particularly, creates table "graph_events" for events and table "graph_event_pairs" for the pairs

Usage

```
createAlignmentTableNew(
  connection,
  trajectoryAnalysisArgs,
  trajectoryLocalArgs,
  g
)
```

Arguments

`g` TrajectoriesGraph object

Value

TrajectoriesGraph object with filled `E(g)$alignedTrajsCount`, `E(g)$alignedTrajsProb` and `V(g)$alignedTrajsCount` values

`createAndFillCohortTable`

Creates and fills cohort table that is going to be used by the Trajectories package

Description

First, it creates the cohort table. Note that this not the standard cohort table that is used in OMOP CDM schema, it is specifically created for this package and later deleted. Second, it loads cohort definition SQL file from the input path (given in `trajectoryAnalysisArgs`) and executes it in the database so the cohort table will be filled in with event-periods that satisfy the cohort requirements. The `cohort_id` of the built cohort is 1.

Usage

```
createAndFillCohortTable(
  connection,
  trajectoryAnalysisArgs,
  trajectoryLocalArgs
)
```

Arguments

`connection` DatabaseConnectorConnection object that is used to connect with database

`trajectoryAnalysisArgs` Object created by `Trajectories:::createTrajectoryAnalysisArgs()` method

`trajectoryLocalArgs` Object created by `Trajectories:::createTrajectoryLocalArgs()` method

`createCohortTable`

Function to create empty cohort table. Script is based on <https://ohdsi.github.io/TheBookOfOhdsi/SqlAndR.html#implementing-the-study-using-sql-and-r> (21th May 2020)

Description

Function to create empty cohort table. Script is based on <https://ohdsi.github.io/TheBookOfOhdsi/SqlAndR.html#implementing-the-study-using-sql-and-r> (21th May 2020)

Usage

```
createCohortTable(connection, trajectoryAnalysisArgs, trajectoryLocalArgs)
```

Arguments

connection DatabaseConnectorConnection object that is used to connect with database

trajectoryAnalysisArgs TrajectoryAnalysisArgs object that must be created by createTrajectoryAnalysisArgs() method

trajectoryLocalArgs TrajectoryLocalArgs object that must be created by createTrajectoryLocalArgs() method

```
createEventPairsTable    Creates event pairs table and populates it with the data
```

Description

Creates event pairs table and populates it with the data

Usage

```
createEventPairsTable(connection, trajectoryAnalysisArgs, trajectoryLocalArgs)
```

Arguments

connection DatabaseConnectorConnection object that is used to connect with database

trajectoryAnalysisArgs Object created by Trajectories:::createTrajectoryAnalysisArgs() method

trajectoryLocalArgs Object created by Trajectories:::createTrajectoryLocalArgs() method

```
createEventPairsTableBrunak  
                                 Creates event pairs table and populates it with the data
```

Description

Creates event pairs table and populates it with the data

Usage

```
createEventPairsTableBrunak(  
  connection,  
  trajectoryAnalysisArgs,  
  trajectoryLocalArgs  
)
```

Arguments

connection	DatabaseConnectorConnection object that is used to connect with database
trajectoryAnalysisArgs	Object created by Trajectories::createTrajectoryAnalysisArgs() method
trajectoryLocalArgs	Object created by Trajectories::createTrajectoryLocalArgs() method

```
createFilteredFullgraphs
```

Creates full graph plots from event pairs (not filtered to specific concept id-s)

Description

Creates full graph plots from event pairs (not filtered to specific concept id-s)

Usage

```
createFilteredFullgraphs(
  connection,
  trajectoryAnalysisArgs,
  trajectoryLocalArgs
)
```

Arguments

connection	Database connection object created by createConnectionDetails() method in DatabaseConnector package
trajectoryAnalysisArgs	TrajectoryAnalysisArgs object that must be created by createTrajectoryAnalysisArgs() method
trajectoryLocalArgs	TrajectoryLocalArgs object that must be created by createTrajectoryLocalArgs() method

```
createTrajectoriesGraph
```

Builds a TrajectoriesGraph object based on event pairs data in "event-PairResultsFilename" file

Description

It is actually an igraph object but has some specific attributes (counts, colors) necessary for Trajectories package

Usage

```
createTrajectoriesGraph(eventPairResultsFilename)
```

Arguments

eventPairResultsFilename
Full path to event pairs file

Details

Vertex attributes: (ID=concept_name), concept_id, count, color, labelcolor

Edge attributes: e1 (name), e1_concept_id, e2 (name), e2_concept_id, e1_count, effect, prob, num-cohortExact. numcohortExact is a number of event periods that had E1->E2 as immediate order (no intermediate events).

createTrajectoryAnalysisArgs

Creates an object to hold analysis-specific data

Description

Creates an object to hold analysis-specific data

Usage

```
createTrajectoryAnalysisArgs(
  mode = "DISCOVERY",
  minimumDaysBetweenEvents = 1,
  maximumDaysBetweenEvents = 3650,
  minPatientsPerEventPair = 10,
  addConditions = T,
  addObservations = F,
  addProcedures = F,
  addDrugExposures = F,
  addDrugEras = F,
  addBirths = F,
  addDeaths = T,
  daysBeforeIndexDate = Inf,
  RRrangeToSkip = c(0, 1.2),
  cohortName = "My sample cohort",
  description = "",
  eventIdsForGraphs = NA
)
```

Arguments

mode
Indicates whether the analysis is run in DISCOVERY or VALIDATION mode. In VALIDATION mode, the package tries to validate predefined event pairs. In DISCOVERY mode, it tries to identify all directional event pairs from the data.

minimumDaysBetweenEvents
The smallest number of days between two events of the patient that can be considered as event pair. Usually we have used 1 but 0 is also possible. The smaller the number is, the more time the calculation takes.

maximumDaysBetweenEvents	The maximum number of days between two events of the patient that can be considered as event pair. Usually we have not really limited it so we have used 3650 (10 years)
minPatientsPerEventPair	Minimum number of people having event1 -> event2 (directional) progression (satisfying minimumDaysBetweenEvents and maximumDaysBetweenEvents requirements) to be included in analysis. If the value is ≥ 1 , it is considered as the absolute count of event pairs. If the value is less than 1, the value is considered as prevalence among the cohort size. For instance, if you have 1000 persons in the cohort and the value is 0.05, each event pair must occur at least $1000 \times 0.05 = 50$ times. Can be used for limiting analysis to frequent event pairs only. However, it does not affect control group matching and therefore, either the p-value.
addConditions	TRUE/FALSE parameter to indicate whether events from Condition_occurrence table should be included in the analysis
addObservations	TRUE/FALSE parameter to indicate whether events from Condition_occurrence table should be included in the analysis
addProcedures	TRUE/FALSE parameter to indicate whether events from Procedure_occurrence table should be included in the analysis
addDrugExposures	TRUE/FALSE parameter to indicate whether events from Drug_exposure table should be included in the analysis. In most of the cases, prefer using addDrugEras instead as the particular RxNorm codes may differ in various databases (leading to no replication) but drug_era is always on ingredient level (active compound) and it also fills gaps between close events.
addDrugEras	TRUE/FALSE parameter to indicate whether events from Drug_era table should be included in the analysis. NB! use either addDrugEras=T or addDrugExposures=T (not both) as it leads to analysis duplication...
addBirths	TRUE/FALSE parameter to indicate whether births events should be included in the analysis.
addDeaths	TRUE/FALSE parameter to indicate whether events from Death table should be included in the analysis.
daysBeforeIndexDate	0 or any positive number that indicates for how many days before index date of the cohort the events are included in the analysis. In case one wants to include all events before index date, use value Inf
cohortName	Reader-friendly short description of the cohort. Used in graph titles and file names (can contain spaces)
description	This is a placeholder for any description of the study/cohort/analysis. For instance, it would be wise to describe here what kind of cohort is that and what the analysis does.
RRrangeToSkip":	Range of relative risks (RR) that are skipped from the analysis. The minimum value for the range is 0. E.g. RRrangeToSkip=c(0,1) searches for $RR > 1$ only (event pairs where the first event increases the risk of the second event). To skip RR with very small effect, it is recommended to use RRrangeToSkip=c(0,1.1) or even RRrangeToSkip=c(0,1.2) in DISCOVERY mode. In case one is interested in pairs with decreasing risk also, it is recommended to use the range something

like `RRrangeToSkip=c(0.8,1.2)` (analyse all pairs that have `RR<0.8` or `R>=1.2`). If you don't want to skip anything, use `RRrangeToSkip=c(1,1)` (analyses all pairs that have `RR<1` or `RR>=1` - that means, all pairs). The first number of the range should be in the range 0..1 and the second number in range 1..Inf

`packageName` Do not use/edit, this is required by `SqlRender::loadRenderTranslateSql`

`eventIdsForGraph` List of exact concept ID-s of the events that are used to align actual trajectories in the end of analysis. Can be left not defined (NA)

Value

TrajectoryAnalysisArgs object

`createTrajectoryLocalArgs`

Creates an object to hold local database-specific parameters

Description

Creates an object to hold local database-specific parameters

Usage

```
createTrajectoryLocalArgs(
  cdmDatabaseSchema,
  vocabDatabaseSchema,
  resultsSchema,
  oracleTempSchema,
  sqlRole = F,
  prefixForResultTableNames = "",
  inputFolder = system.file("extdata", "RA", package = "Trajectories"),
  mainOutputFolder = getwd(),
  databaseHumanReadableName = "My database"
)
```

Arguments

`cdmDatabaseSchema` Schema containing source data in OMOP CDM format

`vocabDatabaseSchema` Schema containing OMOP vocabulary

`resultsSchema` Schema the user has writing access to (used to write analysis tables into)

`oracleTempSchema` In case you are using oracle, schema for temporary tables need to be specified. A schema where temp tables can be created in Oracle. Otherwise leave it as it is (is not used)

`sqlRole` Role to use in SQL for writing tables in 'resultsSchema'. It should also have access to 'cdmDatabaseSchema' and 'vocabDatabaseSchema'. Set to FALSE (or F) if setting to a specific role is not needed. It should be safe to use F if you have no idea of what the SQL roles mean.

prefixForResultTableNames	Table prefix that is used for all output tables to avoid any collision with existing table names. An empty string is also allowed.
inputFolder	Full path to input folder that contains SQL file for cohort definition (SQL Server format) and optionally also trajectoryAnalysisArgs.json. You can use built-in folders of this package such as: inputFolder=system.file("extdata", "RA", package = "Trajectories") which is also the default value. In case your cohort data already exists in the database and you do not need to build it from scratch, set the value to FALSE.
mainOutputFolder	The output folder path. This is the folder where the final results are produced into. Use full path and do NOT add trailing slash! The folder must already exist. Default value is the default working directory.
databaseHumanReadableName	In the future, it will be added to the titles of the graph to indicate what data is this. Use something short. Currently this parameter is not used.
cohortTableSchema	Schema where cohort table is located
cohortTable	Name of the cohort table in cohortTableSchema

Value

TrajectoryLocalArgs object

createValidationSet	<i>Randomly assigns event pairs from the cohort to validation set by using the given ratio</i>
---------------------	--

Description

Randomly assigns event pairs from the cohort to validation set by using the given ratio

Usage

```
createValidationSet(
  connection,
  trajectoryAnalysisArgs,
  trajectoryLocalArgs,
  size = 0.5
)
```

Arguments

connection	DatabaseConnectorConnection object that is used to connect with database
trajectoryAnalysisArgs	Object created by Trajectories:::createTrajectoryAnalysisArgs() method
trajectoryLocalArgs	Object created by Trajectories:::createTrajectoryLocalArgs() method
size	Ratio for assigning event pairs from the cohort to validation set.

createValidationSetup	<i>Creates setup file for the validation of the results in a separate database</i>
-----------------------	--

Description

Creates setup file for the validation of the results in a separate database

Usage

```
createValidationSetup(trajecoryAnalysisArgs, trajectoryLocalArgs)
```

Arguments

trajectoryAnalysisArgs	TrajectoryAnalysisArgs object that must be created by createTrajectoryAnalysisArgs() method
trajectoryLocalArgs	TrajectoryLocalArgs object that must be created by createTrajectoryLocalArgs() method

dbCleanup	<i>Removes all database tables that were created during the analysis</i>
-----------	--

Description

Removes all database tables that were created during the analysis

Usage

```
dbCleanup(connection, trajectoryAnalysisArgs, trajectoryLocalArgs)
```

Arguments

connection	Database connection object created by createConnectionDetails() method in DatabaseConnector package
trajectoryAnalysisArgs	TrajectoryAnalysisArgs object that must be created by createTrajectoryAnalysisArgs() method
trajectoryLocalArgs	TrajectoryLocalArgs object that must be created by createTrajectoryLocalArgs() method

detectSubgraphs	<i>Title</i>
Description	
Title	
Usage	
detectSubgraphs(all_trajs)	
Arguments	
all_trajs	Must have columns trajectory.str, trajectory.count, length
discover	
<i>The main function to run the Trajectory package in discovery mode - that is, to identify the temporal event sequences in the cohort of your local OMOP CDM</i>	

Description

The main function to run the Trajectory package in discovery mode - that is, to identify the temporal event sequences in the cohort of your local OMOP CDM

Usage

```
discover(
  connection,
  trajectoryLocalArgs,
  createCohort = T,
  validationSetSize = 0,
  createEventPairsTable = T,
  runDiscoveryAnalysis = T,
  forceRecalculationOfAnalysis = F,
  createFilteredFullgraphs = T,
  runTrajectoryAnalysis = T,
  selfValidate = F,
  cleanup = F
)
```

Arguments

connection	Database connection object created by createConnectionDetails() method in DatabaseConnector package
trajectoryLocalArgs	TrajectoryLocalArgs object that must be created by createTrajectoryLocalArgs() method
createCohort	Builds a study cohort in the database. If validationSetSize is set to non-zero, splits the cohort into discovery and validation set.

<code>validationSetSize</code>	If set to non-zero, splits the whole cohort into discovery and (self-)validation set. Is meaningful only if <code>selfValidate=T</code> . Allowed values are in range 0..1.
<code>createEventPairsTable</code>	Builds all event pairs and necessary data tables in the database for the analysis. Also clears all results from the database if they exist.
<code>runDiscoveryAnalysis</code>	Run the actual directionality analysis of all event pairs.
<code>forceRecalculationOfAnalysis</code>	Forces deleting previous results from the database and rerunning the whole discovery analysis. Useful mostly in case something goes wrong and you need to force the recalculation (it is, when debugging). In normal circumstances using F is safe.
<code>createFilteredFullgraphs</code>	Builds graphs based on the results.
<code>runTrajectoryAnalysis</code>	If TRUE, runs the trajectory analysis - puts the actual trajectories to the graph
<code>selfValidate</code>	Normally, set to F/FALSE as it is always better to validate your results in another database. However, if you want to validate your results in your own database, then set <code>selfValidate=T</code> and <code>validationSetSize=some meaningful proportion</code> (for example, 0.5). In such case, the discovery analysis is actually conducted on half of the data and the results are then validated on another half.
<code>cleanup</code>	Drops tables from the database that were created during various stages of the analysis.

`estimatedTimeRemaining`

Returns estimated time remaining

Description

Returns estimated time remaining in a textual manner, based on the current progress percentage

Usage

```
estimatedTimeRemaining(progress_perc = 0.25, starttime = Sys.time())
```

Arguments

<code>progress_perc</code>	The percentage of task progress
<code>starttime</code>	Time when current task was started

fillCohortTable	<i>Function to fill the cohort table.</i>
-----------------	---

Description

Function to fill the cohort table.

Usage

```
fillCohortTable(connection, trajectoryAnalysisArgs, trajectoryLocalArgs)
```

Arguments

connection	DatabaseConnectorConnection object that is used to connect with database
trajectoryAnalysisArgs	Object created by Trajectories:::createTrajectoryAnalysisArgs() method
trajectoryLocalArgs	Object created by Trajectories:::createTrajectoryLocalArgs() method

filterByLift	<i>Filtering edges by lift</i>
--------------	--------------------------------

Description

Filtering edges by lift

Usage

```
filterByLift(g, lift.threshold = 2)
```

Arguments

g	igraph object
lift.threshold	lift.threshold

```
filterIgraphRemoveLowEffectLinksAndOrphanNodes
```

Reduces trajectories igraph object by removing low effect links and orphan nodes

Description

Reduces trajectories igraph object by removing low effect links and orphan nodes

Usage

```
filterIgraphRemoveLowEffectLinksAndOrphanNodes(
  g,
  limitOfLinks = 20,
  edge_param_to_sort_by = "effect"
)
```

Arguments

g	igraph object
limitOfLinks	limitOfLinks
edge_param_to_sort_by	edge_param_to_sort_by

```
filterTrajectoriesGraphCrossingEvent
```

Creates a subgraph of edges that go through EVETNAME. Starts building it from EVENTNAME, adds most probable edge. Then takes both events together, adds most probable edge out from these two etc. Limits to limitOfNodes nodes.

Description

Creates a subgraph of edges that go through EVETNAME. Starts building it from EVENTNAME, adds most probable edge. Then takes both events together, adds most probable edge out from these two etc. Limits to limitOfNodes nodes.

Usage

```
filterTrajectoriesGraphCrossingEvent(
  g,
  eventname = 4283892,
  limitOfNodes = F,
  edge_param_to_sort_by = c("effect", "numcohortExact", "numcohortCustom",
    "effectCount", "prob")
)
```


Arguments

g	igraph object
eventname	name of the event
limitOfNodes	limitOfNodes
edge_param_to_sort_by	edge_param_to_sort_by

getCohortSize	<i>Gets cohort size (number of rows in cohort table of that particular cohort ID)</i>
---------------	---

Description

Gets cohort size (number of rows in cohort table of that particular cohort ID)

Usage

```
getCohortSize(connection, trajectoryAnalysisArgs, trajectoryLocalArgs)
```

Arguments

connection	Database connection object created by createConnectionDetails() method in DatabaseConnector package
trajectoryAnalysisArgs	Object created by Trajectories::createTrajectoryAnalysisArgs() method
trajectoryLocalArgs	Object created by Trajectories::createTrajectoryLocalArgs() method

GetOutputFolder	<i>Returns full path to output folder for the results.</i>
-----------------	--

Description

Basically combines the value of mainOutputFolder, database name, and analysis name to get the output folder. Checks also that the folder exists. If createIfMissing=T, then creates the necessary subfolders under mainOutputFolder.

Usage

```
GetOutputFolder(
  trajectoryLocalArgs,
  trajectoryAnalysisArgs,
  createIfMissing = F
)
```

Arguments

- trajectoryLocalArgs
Object created by Trajectories:::createTrajectoryLocalArgs() method
- trajectoryAnalysisArgs
Object created by Trajectories:::createTrajectoryAnalysisArgs() method
- createIfMissing
If TRUE, then creates necessary folder if missing.

Value

Full output path

InitLogger	<i>Initializes logger for Trajectories package</i>
------------	--

Description

Initializes logger for Trajectories package

Usage

```
InitLogger(logfile, threshold = "INFO")
```

Arguments

- logfile
Full path to log file
- threshold
log_threshold values from "ParallelLogger" package: TRACE, DEBUG, INFO, WARN, ERROR

insertTable	<i>Replica of insertTable, but to handle schema problem https://github.com/OHDSI/SqlRender/issues/258 for SqlRender</i>
-------------	---

Description

Replica of insertTable, but to handle schema problem <https://github.com/OHDSI/SqlRender/issues/258> for SqlRender

Usage

```
insertTable(connection, databaseSchema = NULL, ...)
```

```
is.TrajectoryAnalysisArgs
```

Checks whether the object is of type TrajectoryAnalysisArgs

Description

Checks whether the object is of type TrajectoryAnalysisArgs

Usage

```
is.TrajectoryAnalysisArgs(x)
```

Arguments

x	Any R object
---	--------------

```
is.TrajectoryLocalArgs
```

Checks whether the object is of type TrajectoryLocalArgs

Description

Checks whether the object is of type TrajectoryLocalArgs

Usage

```
is.TrajectoryLocalArgs(x)
```

Arguments

x	Any R object
---	--------------

```
IsValidatationMode
```

Returns TRUE if the package is run in validation mode.

Description

Returns TRUE if the package is run in validation mode.

Usage

```
IsValidatationMode(trajjectoryAnalysisArgs, verbose = F)
```

Arguments

trajjectoryAnalysisArgs	Object created by Trajectories:::createTrajectoryAnalysisArgs() method
verbose	If TRUE, outputs some info in INFO/DEBUG log level. Otherwise, returns the results silently.

```
loadRenderTranslateSql
```

Copy of SqlRender::loadRenderTranslateSql() method to correctly solve pathToSql inside Trajectories package

Description

Copy of SqlRender::loadRenderTranslateSql() method to correctly solve pathToSql inside Trajectories package

Usage

```
loadRenderTranslateSql(
  sqlFilename,
  packageName,
  dbms = "sql server",
  ...,
  oracleTempSchema = NULL,
  warnOnMissingParameters = TRUE
)
```

Arguments

warnOnMissingParameters

plotTrajectoriesGraph *Plot TrajectoriesGraph object to PDF file*

Description

Plot TrajectoriesGraph object to PDF file

Usage

```
plotTrajectoriesGraph(
  g,
  layout = igraph::layout_nicely(g),
  outputPdfFullpath = F,
  nodesizes = igraph::V(g)$count,
  linknumbers = igraph::E(g)$numcohortExact,
  linklabels = NA,
  title = ""
)
```

Arguments

g	TrajectoriesGraph object
layout	layout of an igraph object
outputPdfFullpath	Full path to output PDF file
nodesizes	Which values to use for node sizes. By default, uses count values of the nodes from TrajectoriesGraph object
linknumbers	Which numbers to use when calculating the width of on edges. By default, uses numcohortExact values of the edges from TrajectoriesGraph object
linklabels	Which numbers to show on edges. By default, uses linknumbers. Set to NA to use the default value.
title	Title of the graph

PlotTrajectoriesGraphForEvent

Plots graph for a specific event

Description

Plots graph for a specific event

Usage

```
PlotTrajectoriesGraphForEvent(
  connection,
  trajectoryAnalysisArgs,
  trajectoryLocalArgs,
  g,
  eventId = 443732,
  limitOfNodes = 30,
  skipOutputTables = T
)
```

Arguments

connection	Database connection object created by createConnectionDetails() method in DatabaseConnector package
trajectoryAnalysisArgs	TrajectoryAnalysisArgs object that must be created by createTrajectoryAnalysisArgs() method
trajectoryLocalArgs	TrajectoryLocalArgs object that must be created by createTrajectoryLocalArgs() method
g	igraph object
eventId	ID of the event in the event pair
limitOfNodes	limit of the number of nodes to include on the graph
skipOutputTables	If set to T, no output data tables are made (the PDF graphs only).

PlotTrajectoriesGraphForEvents

Creates plots for the analysis results obtained by runEventPairAnalysis()

Description

Analysis results, obtained by runEventPairAnalysis(), have to exist in output folder, set by Trajectories::GetOutputFolder(trajjectoryLocalArgs,trajectoryAnalysisArgs). This function also alignes actual trajectories to the graph, therefore it requires database connection. If you provide an event-Name (OMOP concept name) or list of event names as input, the graph that crosses that event is shown. If you do not provide it (eventName=NA), the function automatically takes most prevalent top5 events and builds graph for each of them.

Usage

```
PlotTrajectoriesGraphForEvents(
  connection,
  trajectoryAnalysisArgs,
  trajectoryLocalArgs,
  eventIds = NA,
  limitOfNodes = 30,
  skipOutputTables = T
)
```

Arguments

- | | |
|------------------------|---|
| connection | Database connection object created by createConnectionDetails() method in DatabaseConnector package |
| trajectoryAnalysisArgs | TrajectoryAnalysisArgs object that must be created by createTrajectoryAnalysisArgs() method |
| trajectoryLocalArgs | TrajectoryLocalArgs object that must be created by createTrajectoryLocalArgs() method |
| eventIds | Exact concept id-s as a list that is used for analyzing/plotting trajectories. If not specified (NA) (recommended) creates trajectories for top 5 events. |
| limitOfNodes | Max number of events in the constructed graph. Used to limit the analysis to most important events only. |
| skipOutputTables | If set to TRUE, no output data tables are made (the PDF graphs only). |

removeOrphanNodes	<i>Removes orphan nodes from igraph/TrajectoriesGraph object</i>
-------------------	--

Description

Removes orphan nodes from igraph/TrajectoriesGraph object

Usage

```
removeOrphanNodes(g)
```

Arguments

g	igraph object
---	---------------

runDiscoveryAnalysis	<i>Runs the analysis that detects statistically significant directional event pairs and writes the results to file. Data is taken from database and it is expected that the tables are created by function createEventPairsTable()</i>
----------------------	--

Description

Runs the analysis that detects statistically significant directional event pairs and writes the results to file. Data is taken from database and it is expected that the tables are created by function createEventPairsTable()

Usage

```
runDiscoveryAnalysis(
  connection,
  trajectoryAnalysisArgs,
  trajectoryLocalArgs,
  forceRecalculation = F
)
```

Arguments

connection	DatabaseConnectorConnection object that is used to connect with database
trajectoryAnalysisArgs	TrajectoryAnalysisArgs object that must be created by createTrajectoryAnalysisArgs() method
trajectoryLocalArgs	TrajectoryLocalArgs object that must be created by createTrajectoryLocalArgs() method
forceRecalculation	Set to TRUE if you wish to recalculate p-values for all pairs again. If it is set to FALSE, it avoids overcalculating p-values for pairs that have been analyzed already.

runValidationAnalysis	<i>Runs the analysis that validates given event pairs and writes the results to file. Data is taken from database and it is expected that the tables are created by function createEventPairsTable()</i>
-----------------------	--

Description

Runs the analysis that validates given event pairs and writes the results to file. Data is taken from database and it is expected that the tables are created by function createEventPairsTable()

Usage

```
runValidationAnalysis(
  connection,
  trajectoryAnalysisArgs,
  trajectoryLocalArgs,
  forceRecalculation = F
)
```

Arguments

connection	DatabaseConnectorConnection object that is used to connect with database
trajectoryAnalysisArgs	TrajectoryAnalysisArgs object that must be created by createTrajectoryAnalysisArgs() method
trajectoryLocalArgs	TrajectoryLocalArgs object that must be created by createTrajectoryLocalArgs() method
forceRecalculation	Set to TRUE if you wish to recalculate p-values for all pairs again. If it is set to FALSE, it avoids overcalculating p-values for pairs that have been analyzed already.
minRelativeRiskToValidate	Relative risk that is used as a minimum threshold for filtering pairs from DISCOVERY STUDY that are being validated

setRole	<i>This function switches role of the database session by simply executing SET ROLE ...;</i>
---------	--

Description

This function switches role of the database session by simply executing SET ROLE ...;

Usage

```
setRole(connection, sqlRole)
```


Arguments

connection	DatabaseConnectorConnection object that is used to connect with database
sqlRole	SQL role to be set

TrajectoryAnalysisArgsFromInputFolder

Searches for trajectoryAnalysisArgs.json file from inputFolder (defined in trajectoryLocalArgs), creates trajectoryAnalysisArgs object from it and returns it.

Description

Searches for trajectoryAnalysisArgs.json file from inputFolder (defined in trajectoryLocalArgs), creates trajectoryAnalysisArgs object from it and returns it.

Usage

TrajectoryAnalysisArgsFromInputFolder(trajectoryLocalArgs)

Arguments

trajectoryLocalArgs	Object created by Trajectories:::createTrajectoryLocalArgs() method
---------------------	---

Value

TrajectoryLocalArgs object

TrajectoryAnalysisArgsFromJson

Reads trajectoryAnalysisArgs object from JSON file

Description

Reads trajectoryAnalysisArgs object from JSON file

Usage

TrajectoryAnalysisArgsFromJson(filepath)

Arguments

filepath	Full path to JSON file
----------	------------------------

Value

TrajectoryAnalysisArgs object

TrajectoryAnalysisArgsToJson

Writes trajectoryAnalysisArgs object to JSON file

Description

Writes trajectoryAnalysisArgs object to JSON file

Usage

```
TrajectoryAnalysisArgsToJson(trajectoryAnalysisArgs, filepath)
```

Arguments

trajectoryAnalysisArgs	Object created by Trajectories:::createTrajectoryAnalysisArgs() method
filepath	Full path to the output file. Should have .json extension as this is actually a JSON file.

validate

The main function to run the Trajectory package in validation mode - that is, to validate someone's results in your local OMOP CDM

Description

The main function to run the Trajectory package in validation mode - that is, to validate someone's results in your local OMOP CDM

Usage

```
validate(
  connection,
  trajectoryLocalArgs,
  createCohort = T,
  createEventPairsTable = T,
  runValidationAnalysis = T,
  forceRecalculationOfAnalysis = F,
  createFilteredFullgraphs = T,
  runTrajectoryAnalysis = F,
  cleanup = T
)
```

Arguments

connection	Database connection object created by createConnectionDetails() method in DatabaseConnector package
trajectoryLocalArgs	TrajectoryLocalArgs object that must be created by createTrajectoryLocalArgs() method

createCohort	Builds a study cohort in the database. Normally set to T/TRUE but for debugging or running the analysis step-by-step one can set it to F/FALSE as well.
createEventPairsTable	Builds all event pairs and necessary data tables in the database for the analysis. Normally set to T/TRUE but for debugging or running the analysis step-by-step one can set it to F/FALSE as well.
runValidationAnalysis	Run the actual directionality analysis of all event pairs. Normally set to T/TRUE but for debugging or running the analysis step-by-step one can set it to F/FALSE as well.
forceRecalculationOfAnalysis	Used only when runValidationAnalysis=T. If TRUE, forces deleting previous results from the database and rerunning the whole validation analysis. Useful mostly in case something goes wrong and you need to force the recalculation (it is, when debugging). In normal circumstances using FALSE is safe.
createFilteredFullgraphs	Builds graphs based on the results. Normally set to T/TRUE but for debugging or running the analysis step-by-step one can set it to F/FALSE as well.
runTrajectoryAnalysis	If TRUE, runs the trajectory analysis - puts the actual trajectories to the graph
cleanup	Drops tables from the database that were created during various stages of the analysis. Normally set to T/TRUE but for debugging or running the analysis step-by-step one can set it to F/FALSE as well.

visualize_data_pairs	<i>Visualize data pairs using Shiny application</i>
----------------------	---

Description

Visualize data pairs using Shiny application

Usage

```
visualize_data_pairs(data)
```

Arguments

data	Data in xlsx format used for visualization. Has to contain columns E1_CONCEPT_ID and E2_CONCEPT_ID
------	--

Examples

```
data <- as.data.frame(read.xlsx("inst/shiny/VisualisingTrajectories/Data/event_pairs_tested.xlsx"))
Trajectories::visualize_data_pairs(data)
```

Index

`addBirthsChecker`, [2](#)
`alignActualTrajectoriesToGraph`, [3](#)
`alignActualTrajectoriesToGraphFull`, [4](#)

`createAlignmentTableNew`, [4](#)
`createAndFillCohortTable`, [5](#)
`createCohortTable`, [5](#)
`createEventPairsTable`, [6](#)
`createEventPairsTableBrunak`, [6](#)
`createFilteredFullgraphs`, [7](#)
`createTrajectoriesGraph`, [7](#)
`createTrajectoryAnalysisArgs`, [8](#)
`createTrajectoryLocalArgs`, [10](#)
`createValidationSet`, [11](#)
`createValidationSetup`, [12](#)

`dbCleanup`, [12](#)
`detectSubgraphs`, [13](#)
`discover`, [13](#)

`estimatedTimeRemaining`, [14](#)

`fillCohortTable`, [15](#)
`filterByLift`, [15](#)
`filterIgraphRemoveLowEffectLinksAndOrphanNodes`,
[16](#)
`filterTrajectoriesGraphCrossingEvent`,
[16](#)

`getCohortSize`, [17](#)
`GetOutputFolder`, [17](#)

`InitLogger`, [18](#)
`insertTable`, [18](#)
`is.TrajectoryAnalysisArgs`, [19](#)
`is.TrajectoryLocalArgs`, [19](#)
`IsValidationMode`, [19](#)

`loadRenderTranslateSql`, [20](#)

`plotTrajectoriesGraph`, [20](#)
`PlotTrajectoriesGraphForEvent`, [21](#)
`PlotTrajectoriesGraphForEvents`, [22](#)

`removeOrphanNodes`, [23](#)

`runDiscoveryAnalysis`, [23](#)
`runValidationAnalysis`, [24](#)

`setRole`, [24](#)

`TrajectoryAnalysisArgsFromInputFolder`,
[25](#)
`TrajectoryAnalysisArgsFromJson`, [25](#)
`TrajectoryAnalysisArgsToJson`, [26](#)

`validate`, [26](#)
`visualize_data_pairs`, [27](#)