Package 'Cohort2Trajectory'

May 6, 2022

Type Package

Title Package for creating patient orientated treatment trajectories from cohorts defined in OHDSI ATLAS

Version 1.0

Date 2022-01-11

Author Markus Haug

Maintainer Markus Haug <markus.haug@ut.ee>

Description Package Cohort2Trajectory creates patient orientated treatment trajectories from cohorts defined in OHDSI ATLAS. The package can be used with access to OMOP CDM database. The package creates discrete and continuous time trajectories (and outputs them as .csv file) which describe patients' treatment through time. Package can be run with GUI or CLI.

License GPL (>= 2)

Imports Rcpp (>= 1.0.7),
shiny,
shinydashboard,
shinycssloaders,
DT,
stringi,
dplyr,
RColorBrewer,
ggplot2,
plyr,
utils,
sortable,
RJSONIO

Remotes OHDSI/CirceR,
OHDSI/CohortGenerator,
OHDSI/DatabaseConnector,
OHDSI/SqlRender,
OHDSI/ParallelLogger,
OHDSI/Eunomia,
OHDSI/ROhdsiWebApi

Suggests testthat (>= 3.0.0),
pheatmap

LinkingTo Rcpp

RoxygenNote 7.1.2

Config/testthat/edition 3

R topics documented:

Cohort2Trajectory-package

Package for creating patient orientated treatment trajectories from cohorts defined in OHDSI ATLAS

Description

Package Cohort2Trajectory creates patient orientated treatment trajectories from cohorts defined in OHDSI ATLAS. The package can be used with access to OMOP CDM database. The package creates discrete and continuous time trajectories (and outputs them as .csv file) which describe patients' treatment through time. Package can be run with GUI or CLI.

Details

This package can be used in two ways 1) Running with GUI 2) Running with CLI

Author(s)

Markus Haug, haugmarkus@protonmail.com.

Maintainer: Markus Haug haugmarkus@protonmail.com

References

This optional section can contain literature or other references for background information.

See Also

HealthInformaticsUT/TrajectoryMarkovAnalysis, HealthInformaticsUT/TrajectoryViz

Examples

```
  ## Not run:
    ## Optional simple examples of the most important functions
    ## These can be in \dontrun{} and \donttest{} blocks.

## End(Not run)
```

Cohort2Trajectory *This function creates patient treatment trajectories*

Description

This function creates patient treatment trajectories

Usage

```
Cohort2Trajectory(
  dbms = "postgresql",
  connection,
  cdmSchema = "ohdsi_cdm",
  cdmTmpSchema = "ohdsi_temp",
  cdmResultsSchema = "ohdsi_results",
  studyName = "Cohort2Trajectory",
  baseUrl = "http://localhost:8080/WebAPI",
  atlasTargetCohort,
  atlasStateCohorts,
  stateCohortLabels,
  stateCohortPriorityOrder,
  stateCohortMandatory,
  stateCohortAbsorbing,
  stateSelectionType,
  trajectoryType,
  lengthOfStay,
  outOfCohortAllowed,
  runSavedStudy = FALSE,
  pathToResults = getwd()
)
```

Arguments

`dbms` The type of DBMS running on the server. Valid values are: 'oracle','postgresql','redshift','sql server','pdw', 'netezza','bigquery','sqlite', 'sqlite extended','spark'

`connection` Connection to database

`cdmSchema` Schema which contains the OHDSI Common Data Model.

`cdmTmpSchema` Schema for temporary tables, will be deleted.

`cdmResultsSchema`
Schema which has the information about the cohorts created in Atlas

`atlasTargetCohort`
The id of the target cohort defined in OHDSI tool ATLAS

`atlasStateCohorts`
The ids of the state cohorts defined in OHDSI tool ATLAS

`stateCohortLabels`
Vector of the customized labels of the state cohorts

`stateCohortPriorityOrder`
Vector of the customized labels of the state cohorts in priority order

`stateCohortMandatory`
Vector of the customized labels of the state cohorts which are mandatory in trajectory

`stateCohortAbsorbing`
Vector of the customized labels of the state cohorts which are absorbing

`stateSelectionType`
The type of state selection (1 - First occurring, 2 - Max overlap, 3 - Priority)

`trajectoryType` The type of the trajectory (0 - Discrete time, 1 - Continuous time)

`lengthOfStay` The length of stay (days) in one state (Effect only in discrete case)

`outOfCohortAllowed`
boolean whether the patient trajectory can surpass the target cohort's observation-period

Examples

```
################################################################################
#
# Example on Eunomia
#
################################################################################

studyName = "TestCohort2Trajectory" #TODO
pathToResults <<-getwd() #TODO

################################################################################
#
# Database credentials
#
################################################################################
pathToDriver = './Drivers'
dbms <- "sqlite" #TODO

cdmSchema = "main" #TODO # Schema which contains the OHDSI Common Data Model
cdmTmpSchema = "main" #TODO # Schema for temporary tables, will be deleted # should be ohdsi_temp
cdmResultsSchema = "main" #TODO # Schema which will contain the final results

baseUrl = NULL #TODO # WebAPI URL is not needed when jsons' are already imported
```

```
################################################################################
#
# Initiate the database connection
#
################################################################################
connectionDetails <- Eunomia::getEunomiaConnectionDetails()
conn <- DatabaseConnector::connect(connectionDetails)
Eunomia::createCohorts(connectionDetails)


################################################################################
#
# Run the study
#
################################################################################

Cohort2Trajectory::Cohort2Trajectory(
  dbms = dbms,
  connection = conn,
  cdmSchema = cdmSchema,
  cdmTmpSchema = cdmTmpSchema,
  cdmResultsSchema = cdmResultsSchema,
  studyName = studyName,
  baseUrl = baseUrl,
  atlasTargetCohort = 1,
  atlasStateCohorts = c(3,4),
  stateCohortLabels = c(”TestState1”, ”TestState2”),
  stateCohortPriorityOrder = c(”TestState1”, ”TestState2”),
  stateCohortMandatory = c(”TestState2”),
  stateCohortAbsorbing = c(”TestState2”),
  ##############################################################################
  # stateSelectionTypes
  # 1 - First occurring
  # 2 - Largest overlap
  # 3 - Priority ordering
  ##############################################################################
  stateSelectionType = 3,
  ##############################################################################
  # trajectoryType
  # 0 - Discrete time
  # 1 - Continuous time
  ##############################################################################
  trajectoryType = 0,
  lengthOfStay = 30,
  outOfCohortAllowed = TRUE,
  runSavedStudy = FALSE,
  pathToResults = pathToResults
)
```

runGUI — *This function initiates the connection with database and starts Shiny application*

Description

This function initiates the connection with database and starts Shiny application

Usage

```
runGUI(
  conn,
  connectionDetails,
  pathToDriver = "./Drivers",
  pathToResults = NULL,
  dbms = "postgresql",
  cdmSchema = "ohdsi_cdm",
  cdmTmpSchema = "ohdsi_temp",
  cdmResultsSchema = "ohdsi_results",
  studyName = "Cohort2Trajectory",
  baseUrl = "http://localhost:8080/WebAPI"
)
```

Arguments

`pathToDriver` Path to a folder containing the JDBC driver JAR files. See downloadJdbcDrivers for instructions on how to download the relevant drivers.

`pathToResults` Path to target directory where results will be saved

`dbms` The type of DBMS running on the server. Valid values are: ’oracle’,’postgresql’,’redshift’,’sql server’,’pdw’, ’netezza’,’bigquery’,’sqlite’, ’sqlite extended’,’spark’

`cdmSchema` Schema which contains the OHDSI Common Data Model.

`cdmTmpSchema` Schema for temporary tables

`cdmResultsSchema` Schema which has the information about the cohorts created in Atlas

`user` The user name used to access the server.

`pw` The password for that user.

`server` The name of the server.

`port` The port on the server to connect to.

Examples

```
################################################################################
#
# Example on Eunomia
#
################################################################################
```

```
studyName = "TestCohort2Trajectory" # TODO
pathToResults <- getwd() # TODO # default value: paste(getwd(), "/tmp", sep = "")

################################################################################
#
# Database credentials
#
################################################################################
pathToDriver = './Drivers'
dbms <- "sqlite" #TODO

cdmSchema = "main" #TODO # Schema which contains the OHDSI Common Data Model
cdmTmpSchema = "main" #TODO # Schema for temporary tables, will be deleted # should be ohdsi_temp
cdmResultsSchema = "main" #TODO # Schema which will contain the final results

baseUrl = NULL #TODO # WebAPI URL is not needed when jsons' are already imported


################################################################################
#
# Initiate the database connection
#
################################################################################
connectionDetails <- Eunomia::getEunomiaConnectionDetails()
conn <- DatabaseConnector::connect(connectionDetails)
Eunomia::createCohorts(connectionDetails)

Cohort2Trajectory::runGUI(
  conn,
  connectionDetails,
  pathToDriver = pathToDriver,
  pathToResults = pathToResults,
  dbms = dbms,
  cdmSchema = cdmSchema,
  cdmTmpSchema = cdmTmpSchema,
  cdmResultsSchema = cdmResultsSchema,
  studyName = studyName,
  baseUrl = baseUrl
)
```

Index