

Package ‘CohortExplorer’

January 20, 2025

Type Package

Title Explorer of Profiles of Patients in a Cohort

Version 0.1.0

Date 2023-10-22

Maintainer Gowtham Rao <rao@ohdsi.org>

Description This software tool is designed to extract data from a randomized subset of individuals within a cohort and make it available for exploration in a shiny application environment. It retrieves date-stamped, event-level records from one or more data sources that represent patient data in the Observational Medical Outcomes Partnership (OMOP) data model format. This tool features a user-friendly interface that enables users to efficiently explore the extracted profiles, thereby facilitating applications, such as reviewing structured profiles.

Depends DatabaseConnector (>= 5.0.0), R (>= 4.0.0)

Imports checkmate, dplyr, lifecycle, ParallelLogger, rlang, stats

Suggests rmarkdown, testthat, knitr, withr

License Apache License

RoxygenNote 7.2.3

VignetteBuilder knitr

Encoding UTF-8

Language en-US

URL <https://ohdsi.github.io/CohortExplorer/>,
<https://github.com/OHDSI/CohortExplorer>

BugReports <https://github.com/OHDSI/CohortExplorer/issues>

NeedsCompilation no

Author Gowtham Rao [aut, cre],
Observational Health Data Science and Informatics [cph]

Repository CRAN

Date/Publication 2023-10-23 15:50:24 UTC

Contents

createCohortExplorerApp	2
exportCohortExplorerAppFiles	4

Index	6
--------------	----------

createCohortExplorerApp

Create Cohort explorer shiny app with person level data

Description

Export person level data from OMOP CDM tables for eligible persons in the cohort. Creates a folder with files that are part of the Cohort Explorer 'shiny' app. This app may then be run to review person level profiles.

Usage

```
createCohortExplorerApp(
  connectionDetails = NULL,
  connection = NULL,
  cohortDatabaseSchema = NULL,
  cdmDatabaseSchema,
  vocabularyDatabaseSchema = cdmDatabaseSchema,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  cohortTable = "cohort",
  cohortDefinitionId,
  cohortName = NULL,
  doNotExportCohortData = FALSE,
  sampleSize = 25,
  personIds = NULL,
  featureCohortDatabaseSchema = NULL,
  featureCohortTable = NULL,
  featureCohortDefinitionSet = NULL,
  exportFolder,
  databaseId,
  shiftDates = FALSE,
  assignNewId = FALSE
)
```

Arguments

connectionDetails

An object of type connectionDetails as created using the [createConnectionDetails](#) function in the DatabaseConnector package. Can be left NULL if connection is provided.

connection	An object of type connection as created using the <code>connect</code> function in the DatabaseConnector package. Can be left NULL if connectionDetails is provided, in which case a new connection will be opened at the start of the function, and closed when the function finishes.
cohortDatabaseSchema	Schema name where your cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
cdmDatabaseSchema	Schema name where your OMOP CDM tables with person level data reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
vocabularyDatabaseSchema	Schema name where your OMOP vocabulary tables reside. It maybe the cdmDatabaseSchema. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
tempEmulationSchema	Some database platforms like Oracle and Impala do not truly support temp tables. To emulate temp tables, provide a schema with write privileges where temp tables can be created.
cohortTable	The name of the cohort table.
cohortDefinitionId	The cohort id to extract records.
cohortName	(optional) Cohort Name
doNotExportCohortData	(Optional) Do you want to not export cohort data? If set to true, parameters cohortDefinitionId, cohort, cohortDatabaseSchema, cohortName will be ignored. The persons entire observation period would be considered the cohort. Cohort Name will be 'Observation Period', cohort id will be set to 0.
sampleSize	(Optional, default = 20) The number of persons to randomly sample. Ignored, if personId is given.
personIds	(Optional) An array of personId's to look for in Cohort table and CDM.
featureCohortDatabaseSchema	The CohortDatabaseSchema where the feature cohort table exists.
featureCohortTable	The Cohort table where feature cohorts are instantiated.
featureCohortDefinitionSet	The CohortDefinitionSet object corresponding to the cohorts to be used as features.
exportFolder	The folder where the output will be exported to. If this folder does not exist it will be created.
databaseId	A short string for identifying the database (e.g. 'Synpuf'). This will be displayed in 'shiny' app to toggle between databases. Should not have space or underscore (_).
shiftDates	(Default = FALSE) Do you want to shift dates? This will help further de-identify data. The shift is the process of re calibrating dates such that all persons mi (observation_period_start_date) is 2000-01-01.

assignNewId (Default = FALSE) Do you want to assign a newId for persons. This will replace the personId in the source with a randomly assigned newId.

Value

Returns invisibly the full path of the export folder where the files were created. In this path are the files that are part of the 'shiny' app.

Examples

```
## Not run:
connectionDetails <- createConnectionDetails(
  dbms = "postgresql",
  server = "ohdsi.com",
  port = 5432,
  user = "me",
  password = "secure"
)

createCohortExplorerApp(
  connectionDetails = connectionDetails,
  cohortDefinitionId = 1234
)

## End(Not run)
```

exportCohortExplorerAppFiles

Copy shiny app files

Description

Copy shiny app files.

Usage

```
exportCohortExplorerAppFiles(exportFolder)
```

Arguments

exportFolder The folder where the output will be exported to. If this folder does not exist it will be created.

Value

Nothing is returned to the environment. The function creates the required files for the shiny application in the export folder.

Examples

```
## Not run:  
exportCohortExplorerAppFiles(  
  exportFolder = "output"  
)  
  
## End(Not run)
```

Index

`connect`, [3](#)
`createCohortExplorerApp`, [2](#)
`createConnectionDetails`, [2](#)
`exportCohortExplorerAppFiles`, [4](#)