Package 'Eunomia'

October 12, 2026

Type Package

Title Standard Dataset Manager for Observational Medical Outcomes Partnership Common Data Model Sample Datasets

Version 2.2.0

Date 2026-10-07

Maintainer Anthony Sena <sena@ohdsi.org>

Description Facilitates access to sample datasets from the 'EunomiaDatasets' repository (<https://github.com/ohdsi/EunomiaDatasets>).

License Apache License 2.0

URL https://ohdsi.github.io/Eunomia/, https://github.com/OHDSI/Eunomia

BugReports https://github.com/OHDSI/Eunomia/issues

Imports arrow, CommonDataModel (>= 1.1.0), DBI, readr, rlang, RSQLite

Suggests DatabaseConnector, duckdb, testthat, withr

Encoding UTF-8

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Anthony Sena [aut, cre],
Frank DeFalco [aut],
Martijn Schuemie [aut],
Natthawut Adulyanukosol [aut],
Star Liu [aut],
Adam Black [aut],
Observational Health Data Science and Informatics [cph]

Repository CRAN

Date/Publication 2026-10-12 19:10:40 UTC

Contents

createCohorts 2
downloadEunomiaData 3

exportDataFiles 4
extractLoadData 4
getDatabaseFile 5
getEunomiaConnectionDetails 7
loadDataFiles 7

Index **9**

`createCohorts` *Construct cohorts*

Description

Creates a set of predefined cohorts in a cohort table. WARNING: this will delete all existing cohorts in the table!

Usage

```
createCohorts(
  connectionDetails,
  cdmDatabaseSchema = "main",
  cohortDatabaseSchema = "main",
  cohortTable = "cohort"
)
```

Arguments

`connectionDetails`	The connection details to connect to the (Eunomia) database.
`cdmDatabaseSchema`	Deprecated. The cdm must be created in the main schema.
`cohortDatabaseSchema`	Deprecated. The cohort table will be created in the main schema.
`cohortTable`	Deprecated. Cohort table will be named "cohort".

Value

A data frame listing all created cohorts.

`downloadEunomiaData` *Download Eunomia data files*

Description

Download the Eunomia data files from https://github.com/OHDSI/EunomiaDatasets

Usage

```
downloadEunomiaData(
  datasetName,
  cdmVersion = "5.5",
  pathToData = Sys.getenv("EUNOMIA_DATA_FOLDER"),
  overwrite = FALSE,
  verbose = FALSE
)
```

Arguments

`datasetName`	The data set name as found on https://github.com/OHDSI/EunomiaDatasets. The data set name corresponds to the folder with the data set ZIP files
`cdmVersion`	The OMOP CDM version. This version will appear in the suffix of the data file, for example: <datasetName>_<cdmVersion>.zip. Default: '5.5'
`pathToData`	The path where the Eunomia data is stored on the file system., By default the value of the environment variable "EUNOMIA_DATA_FOLDER" is used.
`overwrite`	Control whether the existing archive file will be overwritten should it already exist.
`verbose`	Provide additional logging details during execution.

Value

Invisibly returns the destination if the download was successful.

Examples

```
## Not run:
downloadEunomiaData("GiBleed")

## End(Not run)
```

`exportDataFiles` *Export data files from a database(sqlite or duckdb)*

Description

Helper function to export data to csv or parquet files from a database file (sqlite or duckdb).

Usage

```
exportDataFiles(
  dbPath,
  outputFolder,
  outputFormat = "csv",
  dbms = "sqlite",
  verbose = FALSE
)
```

Arguments

`dbPath`	The path to the source .sqlite or .duckdb file
`outputFolder`	The path to the export destination directory
`outputFormat`	The output format for the files. Supported formats include csv, parquet.
`dbms`	The file-based database system to use: 'sqlite' (default) or 'duckdb'
`verbose`	Boolean argument controlling verbose debugging output

Value

No return value, called to export to outputFolder.

`extractLoadData` *Extract the Eunomia data files and load into a database Extract files from a .ZIP file and creates a OMOP CDM database that is then stored in the same directory as the .ZIP file.*

Description

Extract the Eunomia data files and load into a database Extract files from a .ZIP file and creates a OMOP CDM database that is then stored in the same directory as the .ZIP file.

Usage

```
extractLoadData(
  from,
  to,
  dbms = "sqlite",
  cdmVersion = "5.5",
  inputFormat = "csv",
  verbose = FALSE
)
```

Arguments

from	The path to the .ZIP file that contains the csv CDM source files
to	The path to the .sqlite or .duckdb file that will be created
dbms	The file based database system to use: 'sqlite' (default) or 'duckdb'
cdmVersion	The version of the OMOP CDM that are represented in the archive files.
inputFormat	The format of the files expected in the archive. (csv or parquet)
verbose	Provide additional logging details during execution.

Value

No return value, called to load archive into a database file.

See Also

downloadEunomiaData

Examples

```
## Not run:
extractLoadData("c:/strategusData/GiBleed_5.5.zip")

## End(Not run)
```

---

getDatabaseFile *Create a copy of a Eunomia dataset*

Description

Creates a copy of a Eunomia database, and returns the path to the new database file. If the dataset does not yet exist on the user's computer it will attempt to download the source data to the the path defined by the EUNOMIA_DATA_FOLDER environment variable.

Usage

```
getDatabaseFile(
  datasetName,
  cdmVersion = "5.5",
  pathToData = Sys.getenv("EUNOMIA_DATA_FOLDER"),
  dbms = "sqlite",
  databaseFile = tempfile(fileext = paste0(".", dbms)),
  inputFormat = "csv",
  verbose = FALSE,
  overwrite = TRUE
)
```

Arguments

datasetName	The data set name as found on https://github.com/OHDSI/EunomiaDatasets. The data set name corresponds to the folder with the data set ZIP files
cdmVersion	The OMOP CDM version. This version will appear in the suffix of the data file, for example: <datasetName>_<cdmVersion>.zip. Default: '5.5'
pathToData	The path where the Eunomia data is stored on the file system., By default the value of the environment variable "EUNOMIA_DATA_FOLDER" is used.
dbms	The database system to use. "sqlite" (default) or "duckdb"
databaseFile	The path where the database file will be copied to. By default, the database will be copied to a temporary folder, and will be deleted at the end of the R session.
inputFormat	The format of the files expected in the archive. (csv or parquet)
verbose	Provide additional logging details during execution
overwrite	Remove and replace an existing data set.

Value

The file path to the new Eunomia dataset copy

Examples

```
## Not run:
conn <- DBI::dbConnect(RSQLite::SQLite(), getDatabaseFile("GiBleed"))
DBI::dbDisconnect(conn)

conn <- DBI::dbConnect(duckdb::duckdb(), getDatabaseFile("GiBleed", dbms = "duckdb"))
DBI::dbDisconnect(conn, shutdown = TRUE)

conn <- DatabaseConnector::connect(dbms = "sqlite", server = getDatabaseFile("GiBleed"))
DatabaseConnector::disconnect(conn)

## End(Not run)
```

`getEunomiaConnectionDetails` *Get Default Eunomia Connection Details*

Description

Creates a copy of the default GiBleed CDM v5.5 Eunomia database, and provides details for connecting to that copy. Function provides backwards compatibility to prior releases of the Eunomia default GiBleed dataset.

Usage

```
getEunomiaConnectionDetails(
  databaseFile = tempfile(fileext = ".sqlite"),
  dbms = "sqlite",
  overwrite = FALSE
)
```

Arguments

`databaseFile`	The path where the database file will be copied to. By default, the database will be copied to a temporary folder, and will be deleted at the end of the R session.
`dbms`	The target dialect, by default "sqlite".
`overwrite`	Control whether the existing archive file will be overwritten should it already exist.

Value

A ConnectionDetails object, to be used with the `DatabaseConnector` package.

`loadDataFiles` *Load data files into a database(sqlite or duckdb)*

Description

Load data from csv or parquet files into a database file (sqlite or duckdb).

Usage

```
loadDataFiles(
  dataPath,
  dbPath,
  inputFormat = "csv",
  cdmVersion = "5.5",
  cdmDatabaseSchema = "main",
```

```
  dbms = "sqlite",
  verbose = FALSE,
  overwrite = FALSE
)
```

Arguments

dataPath	The path to the directory containing CDM source files (csv or parquet)
dbPath	The path to the .sqlite or .duckdb file that will be created
inputFormat	The input format of the files to load. Supported formats include csv, parquet.
cdmVersion	The CDM version to create in the resulting database. Supported versions are 5.3, 5.4, and 5.5
cdmDatabaseSchema	The schema in which to create the CDM tables. Default is main.
dbms	The file-based database system to use: 'sqlite' (default) or 'duckdb'
verbose	Provide additional logging details during execution.
overwrite	Remove and replace an existing data set.

Value

No return value, loads data into database file.

Index

createCohorts, 2

downloadEunomiaData, 3, *5*

exportDataFiles, 4
extractLoadData, 4

getDatabaseFile, 5
getEunomiaConnectionDetails, 7

loadDataFiles, 7