Package 'CohortDiagnostics'

November 22, 2020

Type Package

Title Diagnostics for OHDSI Cohorts

Version 2.1.0

Date 2020-11-22

Maintainer Gowtham Rao <rao@ohdsi.org>

Description
Diagnostics for cohorts that use the OMOP Common Data Model and the OHDSI tools.

Depends DatabaseConnector (¿= 3.0.0),
FeatureExtraction (¿= 3.1.0),
R (¿= 3.5.0)

Imports Andromeda,
checkmate,
digest,
dplyr (¿= 1.0.0),
ggplot2,
ggiraph,
ParallelLogger (¿= 2.0.0),
readr (¿= 1.4.0),
rlang,
RJSONIO,
ROhdsiWebApi (¿= 1.1.2),
SqlRender (¿= 1.6.7),
stringr,
tidyr (¿= 1.0.0),
methods

Suggests shiny,
shinydashboard,
shinyWidgets,
DT,
plotly,
htmltools,
RColorBrewer,
scales,
knitr,
rmarkdown,
Eunomia,
testthat,
RSQLite (¿= 2.2.1),

zip,
CirceR

Remotes ohdsi/Eunomia,
    ohdsi/FeatureExtraction,
    ohdsi/ROhdsiWebApi,
    ohdsi/DatabaseConnector,
    ohdsi/CirceR

License Apache License

VignetteBuilder knitr

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

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

RoxygenNote 7.1.1

Encoding UTF-8

Language en-US

R topics documented:

---

checkInputFileEncoding

Check character encoding of input file

Description

For its input files, CohortDiagnostics only accepts UTF-8 or ASCII character encoding.
This function can be used to check whether a file meets these criteria.

Usage

```
checkInputFileEncoding(fileName)
```

Arguments

fileName   The path to the file to check

Value

Throws an error if the input file does not have the correct encoding.

---

createCohortTable   *Create cohort table(s)*

Description

This function creates an empty cohort table. Optionally, additional empty tables are created to store statistics on the various inclusion criteria.

Usage

```
createCohortTable(
  connectionDetails = NULL,
  connection = NULL,
  cohortDatabaseSchema,
  cohortTable = "cohort",
  createInclusionStatsTables = FALSE,
  resultsDatabaseSchema = cohortDatabaseSchema,
  cohortInclusionTable = paste0(cohortTable, "_inclusion"),
  cohortInclusionResultTable = paste0(cohortTable, "_inclusion_result"),
  cohortInclusionStatsTable = paste0(cohortTable, "_inclusion_stats"),
  cohortSummaryStatsTable = paste0(cohortTable, "_summary_stats")
)
```

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 connect 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 table resides. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.

cohortTable   Name of the cohort table.

createInclusionStatsTables

      Create the four additional tables for storing inclusion rule statistics?

resultsDatabaseSchema

      Schema name where the statistics tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.

cohortInclusionTable

> Name of the inclusion table, one of the tables for storing inclusion rule statistics.

cohortInclusionResultTable

> Name of the inclusion result table, one of the tables for storing inclusion rule statistics.

cohortInclusionStatsTable

> Name of the inclusion stats table, one of the tables for storing inclusion rule statistics.

cohortSummaryStatsTable

> Name of the summary stats table, one of the tables for storing inclusion rule statistics.

---

createResultsDataModel

Create the results data model tables on a database server.

Description

Create the results data model tables on a database server.

Usage

```
createResultsDataModel(connection = NULL, connectionDetails = NULL, schema)
```

Arguments

connection   An object of type connection as created using the connect 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.

connectionDetails

> An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package. Can be left NULL if connection is provided.

schema   The schema on the postgres server where the tables will be created.

Details

Only PostgreSQL servers are supported.

---

getCohortCounts *Count the cohort(s)*

Description

Computes the subject and entry count per cohort

Usage

```
getCohortCounts(
  connectionDetails = NULL,
  connection = NULL,
  cohortDatabaseSchema,
  cohortTable = "cohort",
  cohortIds = c()
)
```

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 `connect` 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 table resides. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.

cohortTable Name of the cohort table.

cohortIds The cohort Id(s) used to reference the cohort in the cohort table. If left empty, all cohorts in the table will be included.

Value

A tibble with cohort counts

---

getResultsDataModelSpecifications
Get specifications for Cohort Diagnostics results data model

Description

Get specifications for Cohort Diagnostics results data model

Usage

```
getResultsDataModelSpecifications()
```

Value

A tibble data frame object with specifications

---

instantiateCohort   *Instantiate a cohort*

Description

This function instantiates the cohort in the cohort table. Optionally, the inclusion rule statistics are computed and stored in the inclusion rule statistics tables described in createCohortTable).

Usage

```
instantiateCohort(
  connectionDetails = NULL,
  connection = NULL,
  cdmDatabaseSchema,
  oracleTempSchema = NULL,
  cohortDatabaseSchema = cdmDatabaseSchema,
  cohortTable = "cohort",
  baseUrl = NULL,
  cohortJson = NULL,
  cohortSql = NULL,
  cohortId = NULL,
  generateInclusionStats = FALSE,
  resultsDatabaseSchema = cohortDatabaseSchema,
  cohortInclusionTable = paste0(cohortTable, "_inclusion"),
  cohortInclusionResultTable = paste0(cohortTable, "_inclusion_result"),
  cohortInclusionStatsTable = paste0(cohortTable, "_inclusion_stats"),
  cohortSummaryStatsTable = paste0(cohortTable, "_summary_stats")
)
```

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 connect 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.

cdmDatabaseSchema

> Schema name where your patient-level data in OMOP CDM format resides. Note that for SQL Server, this should include both the database and schema name, for example 'cdm_data.dbo'.

oracleTempSchema

> Should be used in Oracle to specify a schema where the user has write privileges for storing temporary tables.

cohortDatabaseSchema

> Schema name where your cohort table resides. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.

cohortTable

> Name of the cohort table.

baseUrl

> The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI". Needn't be provided if `cohortJson` and `cohortSql` are provided.

cohortJson

> A character string containing the JSON of a cohort definition. Needn't be provided if `baseUrl` and `cohortId` are provided.

cohortSql

> The OHDSI SQL representation of the same cohort definition. Needn't be provided if `baseUrl` and `cohortId` are provided.

cohortId

> The cohort ID used to reference the cohort in the cohort table.

generateInclusionStats

> Compute and store inclusion rule statistics?

resultsDatabaseSchema

> Schema name where the statistics tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.

cohortInclusionTable

> Name of the inclusion table, one of the tables for storing inclusion rule statistics.

cohortInclusionResultTable

> Name of the inclusion result table, one of the tables for storing inclusion rule statistics.

cohortInclusionStatsTable

> Name of the inclusion stats table, one of the tables for storing inclusion rule statistics.

cohortSummaryStatsTable

> Name of the summary stats table, one of the tables for storing inclusion rule statistics.

---

instantiateCohortSet *Instantiate a set of cohort*

Description

This function instantiates a set of cohort in the cohort table, using definitions that are fetched from a WebApi interface. Optionally, the inclusion rule statistics are computed and stored in the `inclusionStatisticsFolder`.

Usage

```
instantiateCohortSet(
  connectionDetails = NULL,
  connection = NULL,
  cdmDatabaseSchema,
  oracleTempSchema = NULL,
  cohortDatabaseSchema = cdmDatabaseSchema,
  cohortTable = "cohort",
```

```
    cohortIds = NULL,
    packageName = NULL,
    cohortToCreateFile = "settings/CohortsToCreate.csv",
    baseUrl = NULL,
    cohortSetReference = NULL,
    generateInclusionStats = FALSE,
    inclusionStatisticsFolder = NULL,
    createCohortTable = FALSE,
    incremental = FALSE,
    incrementalFolder = NULL
)
```

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 `connect` 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.

cdmDatabaseSchema

Schema name where your patient-level data in OMOP CDM format resides. Note that for SQL Server, this should include both the database and schema name, for example 'cdm_data.dbo'.

oracleTempSchema

Should be used in Oracle to specify a schema where the user has write privileges for storing temporary tables.

cohortDatabaseSchema

Schema name where your cohort table resides. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.

cohortTable   Name of the cohort table.

cohortIds   Optionally, provide a subset of cohort IDs to restrict the construction to.

packageName   The name of the package containing the cohort definitions. Can be left NULL if `baseUrl` and `cohortSetReference` have been specified.

cohortToCreateFile

The location of the cohortToCreate file within the package. Is ignored if `baseUrl` and `cohortSetReference` have been specified. The cohortToCreateFile must be .csv file that is expected to be read into a dataframe object identical to requirements for `cohortSetReference` argument. This csv file is expected to be encoded in either ASCII or UTF-8, if not, an error message will be displayed and process stopped.

baseUrl   The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI". Can be left NULL if `packageName` and `cohortToCreateFile` have been specified.

cohortSetReference

A data frame with four columns, as described in the details. Can be left NULL if `packageName` and `cohortToCreateFile` have been specified.

`generateInclusionStats`
> Compute and store inclusion rule statistics?

`inclusionStatisticsFolder`
> The folder where the inclusion rule statistics are stored. Can be left NULL if `generateInclusionStats = FALSE`.

`createCohortTable`
> Create the cohort table? If `incremental = TRUE` and the table already exists this will be skipped.

`incremental`   Create only cohorts that haven't been created before?

`incrementalFolder`
> If `incremental = TRUE`, specify a folder where records are kept of which definition has been executed.

Details

Currently two ways of executing this function are supported, either (1) [Package Mode] embedded in a study package, assuming the cohort definitions are stored in that package using the `ROhdsiWebApi::insertCohortDefinitionSetInPackage`, or (2) [WebApi Mode] By using a WebApi interface to retrieve the cohort definitions.

When using this function in Package Mode: Use the `packageName` and `cohortToCreateFile` to specify the name of the study package, and the name of the cohortToCreate file within that package, respectively

When using this function in WebApi Mode: use the `baseUrl` and `cohortSetReference` to specify how to connect to the WebApi, and which cohorts to fetch, respectively.

Note: if the parameters for both Package Mode and WebApi Mode are provided, then Package mode is preferred.

The `cohortSetReference` argument must be a data frame with the following columns:

referentConceptId A standard omop concept id that serves as the referant phenotype definition for the cohort Id.

cohortId The cohort Id is the id used to identify a cohort definition. This is required to be unique. It will be used to create file names. It is recommended to be (referrentConceptId * 1000) + a number between 3 to 999

webApiCohortId Cohort Id in the webApi/atlas instance. It is a required field to run Cohort Diagnostics in WebApi mode. It is discarded in package mode.

cohortName The full name of the cohort. This will be shown in the Shiny app.

logicDescription A human understandable brief description of the cohort definition. This logic does not have to a fully specified description of the cohort definition, but should provide enough context to help user understand the meaning of the cohort definition

Value

A data frame with cohort counts

launchCohortExplorer   *Launch the CohortExplorer Shiny app*

Description

Launch the CohortExplorer Shiny app

Usage

```
launchCohortExplorer(
  connectionDetails,
  cdmDatabaseSchema,
  cohortDatabaseSchema,
  cohortTable,
  cohortId,
  sampleSize = 100,
  subjectIds = NULL
)
```

Arguments

connectionDetails

> An object of type `connectionDetails` as created using the `createConnectionDetails` function in the DatabaseConnector package.

cdmDatabaseSchema

> Schema name where your patient-level data in OMOP CDM format resides. Note that for SQL Server, this should include both the database and schema name, for example 'cdm_data.dbo'.

cohortDatabaseSchema

> Schema name where your cohort table resides. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.

cohortTable   Name of the cohort table.

cohortId   The ID of the cohort.

sampleSize   Number of subjects to sample from the cohort. Ignored if subjectIds is specified.

subjectIds   A vector of subject IDs to view.

Details

Launches a Shiny app that allows the user to explore a cohort of interest.

launchDiagnosticsExplorer

Launch the Diagnostics Explorer Shiny app

Description

Launch the Diagnostics Explorer Shiny app

Usage

```
launchDiagnosticsExplorer(
  dataFolder = "data",
  connectionDetails = NULL,
  resultsDatabaseSchema = NULL,
  vocabularyDatabaseSchema = resultsDatabaseSchema,
  aboutText = NULL,
  cohortBaseUrl = "https://atlas.ohdsi.org/#/cohortdefinition/",
  conceptBaseUrl = "https://athena.ohdsi.org/search-terms/terms/",
  runOverNetwork = FALSE,
  port = 80,
  launch.browser = FALSE
)
```

Arguments

dataFolder	A folder where the premerged file is stored. Use the `preMergeDiagnosticsFiles` function to generate this file.
connectionDetails	An object of type `connectionDetails` as created using the `createConnectionDetails` function in the DatabaseConnector package, specifying how to connect to the server where the CohortDiagnostics results have been uploaded using the `uploadResults` function.
resultsDatabaseSchema	The schema on the database server where the CohortDiagnostics results have been uploaded.
vocabularyDatabaseSchema	The schema on the database server where the vocabulary tables are located.
aboutText	Text (using HTML markup) that will be displayed in an About tab in the Shiny app. If not provided, no About tab will be shown.
cohortBaseUrl	The base URL for constructing linkouts to an ATLAS instance, using the webApiCohortId in the cohortsToCreate file. If NULL, no linkouts will be created.
conceptBaseUrl	The base URL for constructing linkouts to an Athena instance, using the concept ID.
runOverNetwork	(optional) Do you want the app to run over your network?
port	(optional) Only used if `runOverNetwork` = TRUE.
launch.browser	Should the app be launched in your default browser, or in a Shiny window. Note: copying to clipboard will not work in a Shiny window.

Details

Launches a Shiny app that allows the user to explore the diagnostics

---

`preMergeDiagnosticsFiles`

Premerge Shiny diagnostics files

Description

This function combines diagnostics results from one or more databases into a single file. The result is a single file that can be used as input for the Diagnostics Explorer Shiny app.

It also checks whether the results conform to the results data model specifications.

Usage

```
preMergeDiagnosticsFiles(dataFolder, tempFolder = tempdir())
```

Arguments

`dataFolder`	folder where the exported zip files for the diagnostics are stored. Use the runCohortDiagnostics function to generate these zip files. Zip files containing results from multiple databases may be placed in the same folder.
`tempFolder`	A folder on the local file system where the zip files are extracted to. Will be cleaned up when the function is finished. Can be used to specify a temp folder on a drive that has sufficient space if the default system temp space is too limited.

`runCohortDiagnostics`   *Run cohort diagnostics*

Description

Runs the cohort diagnostics on all (or a subset of) the cohorts instantiated using the `ROhdsiWebApi::insertCohortDefinitionSetInPackage` function. Assumes the cohorts have already been instantiated.

Characterization: If runTemporalCohortCharacterization argument is TRUE, then the following default covariateSettings object will be created using `RFeatureExtraction::createTemporalCovariateSe` Alternatively, a covariate setting object may be created using the above as an example.

Usage

```
runCohortDiagnostics(
  packageName = NULL,
  cohortToCreateFile = "settings/CohortsToCreate.csv",
  baseUrl = NULL,
  cohortSetReference = NULL,
  phenotypeDescriptionFile = NULL,
  connectionDetails = NULL,
  connection = NULL,
  cdmDatabaseSchema,
  oracleTempSchema = NULL,
  cohortDatabaseSchema,
  cohortTable = "cohort",
  cohortIds = NULL,
  inclusionStatisticsFolder = file.path(exportFolder, "inclusionStatistics"),
  exportFolder,
  databaseId,
  databaseName = databaseId,
  databaseDescription = databaseId,
  cdmVersion = 5,
  runInclusionStatistics = TRUE,
  runIncludedSourceConcepts = TRUE,
  runOrphanConcepts = TRUE,
  runTimeDistributions = TRUE,
  runVisitContext = TRUE,
  runBreakdownIndexEvents = TRUE,
  runIncidenceRate = TRUE,
  runCohortOverlap = TRUE,
  runCohortCharacterization = TRUE,
  covariateSettings = createDefaultCovariateSettings(),
  runTemporalCohortCharacterization = TRUE,
 temporalCovariateSettings = createTemporalCovariateSettings(useConditionOccurrence =
   TRUE, useDrugEraStart = TRUE, useProcedureOccurrence = TRUE, useMeasurement = TRUE,
   temporalStartDays = c(-365, -30, 0, 1, 31), temporalEndDays = c(-31, -1, 0, 30, 365)),
  minCellCount = 5,
  incremental = FALSE,
  incrementalFolder = file.path(exportFolder, "incremental")
)
```

Arguments

packageName   The name of the package containing the cohort definitions. Can be left
               NULL if `baseUrl` and `cohortSetReference` have been specified.

cohortToCreateFile

               The location of the cohortToCreate file within the package. Is ignored if
               `baseUrl` and `cohortSetReference` have been specified. The cohortToCre-
               ateFile must be .csv file that is expected to be read into a dataframe
               object identical to requirements for `cohortSetReference` argument. This
               csv file is expected to be encoded in either ASCII or UTF-8, if not, an
               error message will be displayed and process stopped.

baseUrl   The base URL for the WebApi instance, for example: "http://server.org:80/WebAPI".
               Can be left NULL if `packageName` and `cohortToCreateFile` have been

specified.

cohortSetReference
A data frame with four columns, as described in the details. Can be left NULL if `packageName` and `cohortToCreateFile` have been specified.

phenotypeDescriptionFile
(Optional) The location of the phenotype description file within the package. The file must be .csv file and have the following columns that may be read into following data types: phenotypeId (double), phenotypeName (character), referentConceptId (double), clinicalDescription (character), literatureReview (character), phenotypeNotes (character). Note: the field names are in snake_case. Also, character fields should not have 'NA' in it. 'NA's are commonly added by R when R functions exports data from dataframe into CSV. Instead please use '' (empty string) to represent absence of data. The literature_review field is expected to be a html link to page that contains resources related to literature review for the phenotype, and will be used to create link-out object. This csv file is expected to be encoded in either ASCII or UTF-8, if not, an error message will be displayed and process stopped.

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 `connect` 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.

cdmDatabaseSchema
Schema name where your patient-level data in OMOP CDM format resides. Note that for SQL Server, this should include both the database and schema name, for example 'cdm_data.dbo'.

oracleTempSchema
Should be used in Oracle to specify a schema where the user has write privileges for storing temporary tables.

cohortDatabaseSchema
Schema name where your cohort table resides. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.

cohortTable   Name of the cohort table.

cohortIds   Optionally, provide a subset of cohort IDs to restrict the diagnostics to.

inclusionStatisticsFolder
The folder where the inclusion rule statistics are stored. Can be left NULL if `runInclusionStatistics` = FALSE.

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').

databaseName   The full name of the database. If NULL, defaults to databaseId.

databaseDescription
A short description (several sentences) of the database. If NULL, defaults to databaseId.

cdmVersion   The version of the OMOP CDM. Default 5. (Note: only 5 is supported.)

runInclusionStatistics

        Generate and export statistic on the cohort inclusion rules?

runIncludedSourceConcepts

        Generate and export the source concepts included in the cohorts?

runOrphanConcepts

        Generate and export potential orphan concepts?

runTimeDistributions

        Generate and export cohort time distributions?

runVisitContext

        Generate and export index-date visit context?

runBreakdownIndexEvents

        Generate and export the breakdown of index events?

runIncidenceRate

        Generate and export the cohort incidence rates?

runCohortOverlap

        Generate and export the cohort overlap? Overlaps are checked within cohortIds that have the same phenotype ID sourced from the CohortSetReference or cohortToCreateFile.

runCohortCharacterization

        Generate and export the cohort characterization? Only records with values greater than 0.0001 are returned.

covariateSettings

        Either an object of type covariateSettings as created using one of the createCovariateSettings function in the FeatureExtraction package, or a list of such objects.

runTemporalCohortCharacterization

        Generate and export the temporal cohort characterization? Only records with values greater than 0.001 are returned.

temporalCovariateSettings

        Either an object of type covariateSettings as created using one of the createTemporalCovariateSettings function in the FeatureExtraction package, or a list of such objects.

minCellCount   The minimum cell count for fields contains person counts or fractions.

incremental   Create only cohort diagnostics that haven't been created before?

incrementalFolder

        If incremental = TRUE, specify a folder where records are kept of which cohort diagnostics has been executed.

Details

Currently two ways of executing this function are supported, either (1) [Package Mode] embedded in a study package, assuming the cohort definitions are stored in that package using the ROhdsiWebApi::insertCohortDefinitionSetInPackage, or (2) [WebApi Mode] By using a WebApi interface to retrieve the cohort definitions.

When using this function in Package Mode: Use the packageName and cohortToCreateFile to specify the name of the study package, and the name of the cohortToCreate file within that package, respectively

When using this function in WebApi Mode: use the baseUrl and cohortSetReference to specify how to connect to the WebApi, and which cohorts to fetch, respectively.

Note: if the parameters for both Package Mode and WebApi Mode are provided, then Package mode is preferred.

The cohortSetReference argument must be a data frame with the following columns:

referentConceptId A standard omop concept id that serves as the referant phenotype definition for the cohort Id.

cohortId The cohort Id is the id used to identify a cohort definition. This is required to be unique. It will be used to create file names. It is recommended to be (referentConceptId * 1000) + a number between 3 to 999

webApiCohortId Cohort Id in the webApi/atlas instance. It is a required field to run Cohort Diagnostics in WebApi mode. It is discarded in package mode.

cohortName The full name of the cohort. This will be shown in the Shiny app.

logicDescription A human understandable brief description of the cohort definition. This logic does not have to a fully specified description of the cohort definition, but should provide enough context to help user understand the meaning of the cohort definition

---

uploadPrintFriendly	*Upload print-friendly cohort representations to the database server.*

Description

For all the cohorts in the 'cohort' table, this will generate print-friendly text and store it in a new table called 'cohort_extra'.

Usage

```
uploadPrintFriendly(connectionDetails = NULL, schema)
```

Arguments

connectionDetails

An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package.

schema   The schema on the postgres server where the results are stored.

---

uploadResults	*Upload results to the database server.*

Description

Requires the results data model tables have been created using the createResultsDataModel function.

Set the POSTGRES_PATH environmental variable to the path to the folder containing the psql executable to enable bulk upload (recommended).

Usage

```
uploadResults(
  connectionDetails = NULL,
  schema,
  zipFileName,
  forceOverWriteOfSpecifications = FALSE,
  purgeSiteDataBeforeUploading = TRUE,
  tempFolder = tempdir()
)
```

Arguments

connectionDetails

> An object of type connectionDetails as created using the createConnectionDetails function in the DatabaseConnector package.

schema   The schema on the postgres server where the tables have been created.

zipFileName   The name of the zip file.

forceOverWriteOfSpecifications

> If TRUE, specifications of the phenotypes, cohort definitions, and analysis will be overwritten if they already exist on the database. Only use this if these specifications have changed since the last upload.

purgeSiteDataBeforeUploading

> If TRUE, before inserting data for a specific databaseId all the data for that site will be dropped. This assumes the input zip file contains the full data for that data site.

tempFolder   A folder on the local file system where the zip files are extracted to. Will be cleaned up when the function is finished. Can be used to specify a temp folder on a drive that has sufficent space if the default system temp space is too limited.

Index