

Package ‘CaseControl’

July 14, 2022

Type Package

Title Case-Control

Version 3.2.0

Date 2022-07-14

Maintainer Martijn Schuemie <schuemie@ohdsi.org>

Description CaseControl is an R package for performing (nested) matched case-control analyses in an observational database in the OMOP Common Data Model.

VignetteBuilder knitr

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

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

Depends R (>= 3.2.2),
Cyclops (>= 3.1.1),
DatabaseConnector (>= 4.0.0),
Andromeda,
survival,
FeatureExtraction (>= 3.0.0)

Imports SqlRender (>= 1.7.0),
rlang,
dplyr,
Rcpp (>= 0.11.2),
ParallelLogger (>= 2.0.0),
plyr,
cli,
pillar,
bit64

Suggests testthat,
knitr,
rmarkdown,
EmpiricalCalibration

Remotes ohdsi/FeatureExtraction

License Apache License 2.0

LinkingTo Rcpp

NeedsCompilation yes

RoxygenNote 7.2.0

Encoding UTF-8

R topics documented:

CaseData-class	2
computeMdr	3
createCaseControlData	4
createCcAnalysis	5
createCreateCaseControlDataArgs	6
createExposureOutcomeNestingCohort	6
createFitCaseControlModelArgs	7
createGetDbCaseDataArgs	8
createGetDbExposureDataArgs	9
createMatchingCriteria	9
createSamplingCriteria	10
createSelectControlsArgs	11
createSimpleCovariateSettings	12
fitCaseControlModel	12
getAttritionTable	13
getDbCaseData	14
getDbExposureData	16
insertDbPopulation	17
isCaseData	18
loadCaseControlsExposure	18
loadCaseData	19
loadCcAnalysisList	19
loadExposureOutcomeNestingCohortList	20
runCcAnalyses	20
saveCaseControlsExposure	22
saveCaseData	23
saveCcAnalysisList	23
saveExposureOutcomeNestingCohortList	24
selectControls	24
summarizeCcAnalyses	25
Index	26

CaseData-class

*Case Data***Description**

CaseData is an S4 class that inherits from [Andromeda](#). It contains information on cases.

A CaseData object is typically created using [getDbCaseData](#), can only be saved using [saveCaseData](#), and loaded using [loadCaseData](#).

Usage

```
## S4 method for signature 'CaseData'
show(object)
```

```
## S4 method for signature 'CaseData'
summary(object)
```

Arguments

object An object of class 'CaseData'.

See Also

[isCaseData](#)

computeMdr

Compute the minimum detectable relative risk

Description

Compute the minimum detectable relative risk

Usage

```
computeMdr(caseControlData, alpha = 0.05, power = 0.8, twoSided = TRUE)
```

Arguments

caseControlData

A data frame describing the cases and controls as created using the [createCaseControlData](#) function. This should at least have these columns: isCase, exposed.

alpha

Type I error.

power

1 - beta, where beta is the type II error.

twoSided

Consider a two-sided test?

Details

Compute the minimum detectable relative risk (MDRR) for a given study population, using the actual observed sample size and number of exposed controls. Computations by Miettinen (1969) and Rothman and Boice (1979) are used. Based on and verified using Ken Rothman's EpiSheet.

Value

A data frame with the MDRR and some counts.

References

Miettinen OS (1969) Individual matching in the case of all or none responses. *Biometrics*, 25, 339-354.

Rothman KJ, Boice JD (1979) *Epidemiologic Analysis with a Programmable Calculator*. NIH Publication No.79-1649.

createCaseControlData *Create case-control data*

Description

Create case-control data

Usage

```
createCaseControlData(
  caseControlsExposure,
  exposureId,
  firstExposureOnly = FALSE,
  riskWindowStart = 0,
  riskWindowEnd = 0,
  exposureWashoutPeriod = 0
)
```

Arguments

caseControlsExposure An object of type `caseControlsExposure` as created using the [getDbExposureData](#) function.

exposureId The identifier of the exposure.

firstExposureOnly Should only the first exposure per subject be included?

riskWindowStart The start of the risk window (in days) relative to the index date. This number should be non-positive.

riskWindowEnd The end of the risk window (in days) relative to the index date. This number should be non-positive.

exposureWashoutPeriod Minimum required numbers of days of observation for inclusion of an exposure.

Details

For each case and control, assesses whether exposure takes place within the risk window. The output can be directly used in a conditional logistic regression.

Value

A data frame with these columns:

personId The person ID

indexDate The index date

isCase Is the person a case or a control?

stratumId The ID linking cases and controls in a matched set

exposed Was the subject exposed during the risk window?

createCcAnalysis	<i>Create a case-control analysis specification</i>
------------------	---

Description

Create a case-control analysis specification

Usage

```
createCcAnalysis(
  analysisId = 1,
  description = "",
  exposureType = NULL,
  outcomeType = NULL,
  nestingCohortType = NULL,
  getDbCaseDataArgs,
  selectControlsArgs,
  getDbExposureDataArgs,
  createCaseControlDataArgs,
  fitCaseControlModelArgs
)
```

Arguments

analysisId	An integer that will be used later to refer to this specific set of analysis choices.
description	A short description of the analysis.
exposureType	If more than one exposure is provided for each exposureOutcomeNestingCohort, this field should be used to select the specific exposure to use in this analysis.
outcomeType	If more than one outcome is provided for each exposureOutcomeNestingCohort, this field should be used to select the specific outcome to use in this analysis.
nestingCohortType	If more than one nesting cohort is provided for each exposureOutcomeNestingCohort, this field should be used to select the specific nesting cohort to use in this analysis.
getDbCaseDataArgs	An object representing the arguments to be used when calling the createGetDbCaseDataArgs function.
selectControlsArgs	An object representing the arguments to be used when calling the createSelectControlsArgs function.
getDbExposureDataArgs	An object representing the arguments to be used when calling the createGetDbExposureDataArgs function.
createCaseControlDataArgs	An object representing the arguments to be used when calling the createCreateCaseControlDataArgs function.
fitCaseControlModelArgs	An object representing the arguments to be used when calling the createFitCaseControlModelArgs function.

Details

Create a set of analysis choices, to be used with the [runCcAnalyses](#) function.

```
createCreateCaseControlDataArgs
```

Create a parameter object for the function createCaseControlData

Description

Create a parameter object for the function createCaseControlData

Usage

```
createCreateCaseControlDataArgs(
  firstExposureOnly = FALSE,
  riskWindowStart = 0,
  riskWindowEnd = 0,
  exposureWashoutPeriod = 0
)
```

Arguments

`firstExposureOnly` Should only the first exposure per subject be included?

`riskWindowStart` The start of the risk window (in days) relative to the index date. This number should be non-positive.

`riskWindowEnd` The end of the risk window (in days) relative to the index date. This number should be non-positive.

`exposureWashoutPeriod` Minimum required numbers of days of observation for inclusion of an exposure.

Details

Create an object defining the parameter values.

```
createExposureOutcomeNestingCohort
```

Create exposure-outcome-nesting-cohort combinations.

Description

Create exposure-outcome-nesting-cohort combinations.

Usage

```
createExposureOutcomeNestingCohort(
  exposureId,
  outcomeId,
  nestingCohortId = NULL
)
```

Arguments

exposureId	A concept ID identifying the target drug in the exposure table. If multiple strategies for picking the exposure will be tested in the analysis, a named list of numbers can be provided instead. In the analysis, the name of the number to be used can be specified using the #' exposureType parameter in the createCcAnalysis function.
outcomeId	A concept ID identifying the outcome in the outcome table. If multiple strategies for picking the outcome will be tested in the analysis, a named list of numbers can be provided instead. In the analysis, the name of the number to be used can be specified using the outcomeType parameter in the createCcAnalysis function.
nestingCohortId	A concept ID identifying the nesting cohort in the nesting cohort table. If multiple strategies for picking the nesting cohort will be tested in the analysis, a named list of numbers can be provided instead. In the analysis, the name of the number to be used can be specified using the nestingCohortType parameter in the createCcAnalysis function.

Details

Create a set of hypotheses of interest, to be used with the [runCcAnalyses](#) function.

```
createFitCaseControlModelArgs
```

Create a parameter object for the function fitCaseControlModel

Description

Create a parameter object for the function fitCaseControlModel

Usage

```
createFitCaseControlModelArgs(
  useCovariates = FALSE,
  excludeCovariateIds = c(),
  includeCovariateIds = c(),
  profileGrid = NULL,
  profileBounds = c(log(0.1), log(10)),
  prior = createPrior("laplace", useCrossValidation = TRUE),
  control = createControl(cvType = "auto", startingVariance = 0.01, tolerance = 2e-07,
    cvRepetitions = 10, selectorType = "byPid", noiseLevel = "quiet")
)
```

Arguments

useCovariates	Whether to use the covariates in the caseControlsExposure.
excludeCovariateIds	Exclude these covariates from the model.
includeCovariateIds	Include only these covariates in the model.

profileGrid	A one-dimensional grid of points on the log(relative risk) scale where the likelihood for coefficient of variables is sampled. See details.
profileBounds	The bounds (on the log relative risk scale) for the adaptive sampling of the likelihood function.
prior	The prior used to fit the model. See createPrior for details.
control	The control object used to control the cross-validation used to determine the hyperparameters of the prior (if applicable). See createControl for details.

Details

Create an object defining the parameter values.

```
createGetDbCaseDataArgs
```

Create a parameter object for the function getDbCaseData

Description

Create a parameter object for the function getDbCaseData

Usage

```
createGetDbCaseDataArgs(
  useNestingCohort = FALSE,
  useObservationEndAsNestingEndDate = TRUE,
  getVisits = FALSE,
  studyStartDate = "",
  studyEndDate = "",
  maxNestingCohortSize = 1e+07,
  maxCasesPerOutcome = 5e+05
)
```

Arguments

useNestingCohort	Should the study be nested in a cohort (e.g. people with a specific indication)? If not, the study will be nested in the general population.
useObservationEndAsNestingEndDate	When using a nesting cohort, should the observation period end date be used instead of the cohort end date?
getVisits	Get data on visits? This is needed when matching on visit date is requested later on.
studyStartDate	A calendar date specifying the minimum date where data is used. Date format is 'yyyymmdd'.
studyEndDate	A calendar date specifying the maximum date where data is used. Date format is 'yyyymmdd'.
maxNestingCohortSize	If the nesting cohort is larger than this number it will be sampled to this size. maxCohortSize = 0 indicates no maximum size.
maxCasesPerOutcome	If there are more than this number of cases for a single outcome cases will be sampled to this size. maxCasesPerOutcome = 0 indicates no maximum size.

Details

Create an object defining the parameter values.

```
createGetDbExposureDataArgs
```

Create a parameter object for the function getDbExposureData

Description

Create a parameter object for the function getDbExposureData

Usage

```
createGetDbExposureDataArgs(covariateSettings = NULL)
```

Arguments

covariateSettings

Either an object of type covariateSettings as created using the createCovariateSettings function in the FeatureExtraction package, or an object of type SimpleCovariateSettings as created using the createSimpleCovariateSettings function. If NULL then no covariate data is retrieved.

Details

Create an object defining the parameter values.

```
createMatchingCriteria
```

Create matching criteria

Description

Criteria to use to select individual matches for cases.

Usage

```
createMatchingCriteria(  
  controlsPerCase = 1,  
  matchOnAge = TRUE,  
  ageCaliper = 2,  
  matchOnGender = TRUE,  
  matchOnProvider = FALSE,  
  matchOnCareSite = FALSE,  
  matchOnVisitDate = FALSE,  
  visitDateCaliper = 30,  
  matchOnTimeInCohort = FALSE,  
  daysInCohortCaliper = 30,  
  removedUnmatchedCases = TRUE,  
  seed = 1  
)
```

Arguments

controlsPerCase	Maximum number of controls to select per case.
matchOnAge	Match on age?
ageCaliper	Maximum difference (in years) in age when matching on age.
matchOnGender	Match on gender?
matchOnProvider	Match on provider (as specified in the person table)?
matchOnCareSite	Match on care site (as specified in the person table)?
matchOnVisitDate	Should the index date of the control be changed to the nearest visit date?
visitDateCaliper	Maximum difference (in days) between the index date and the visit date when matching on visit date.
matchOnTimeInCohort	Match on time in nesting cohort? When not using nesting, this is interpreted as time observed prior to index.
daysInCohortCaliper	Maximum difference (in days) in time in cohort.
removedUnmatchedCases	Should cases with no matched controls be removed?
seed	The number generator seed. A null value sets seed via Sys.time .

Value

A settings object to be used in the [selectControls](#) function.

```
createSamplingCriteria
```

Create sampling criteria

Description

Criteria to use when controls are simply sampled from the (nesting) population.

Usage

```
createSamplingCriteria(controlsPerCase = 1, seed = 1)
```

Arguments

controlsPerCase	Maximum number of controls to select per case.
seed	The number generator seed. A null value sets seed via Sys.time .

Value

A settings object to be used in the [selectControls](#) function.

`createSelectControlsArgs`*Create a parameter object for the function selectControls*

Description

Create a parameter object for the function selectControls

Usage

```
createSelectControlsArgs(  
  firstOutcomeOnly = TRUE,  
  washoutPeriod = 180,  
  controlSelectionCriteria = createMatchingCriteria(),  
  minAge = NULL,  
  maxAge = NULL  
)
```

Arguments

<code>firstOutcomeOnly</code>	Use the first outcome per person?
<code>washoutPeriod</code>	Minimum required numbers of days of observation for inclusion as either case or control.
<code>controlSelectionCriteria</code>	Either a <code>matchingCriteria</code> object as generated using the <code>createMatchingCriteria</code> function, or a <code>samplingCriteria</code> object as generated using the <code>createSamplingCriteria</code> function.
<code>minAge</code>	Minimum age at which patient time will be included in the analysis. Note that information prior to the min age is still used to determine exposure status after the minimum age (e.g. when a prescription was started just prior to reaching the minimum age). Also, outcomes occurring before the minimum age is reached will be considered as prior outcomes when using first outcomes only. Age should be specified in years, but non-integer values are allowed. If not specified, no age restriction will be applied.
<code>maxAge</code>	Maximum age at which patient time will be included in the analysis. Age should be specified in years, but non-integer values are allowed. If not specified, no age restriction will be applied.

Details

Create an object defining the parameter values.

```
createSimpleCovariateSettings
    Create simple covariate settings
```

Description

Create simple covariate settings

Usage

```
createSimpleCovariateSettings(
  useDemographicsAgeGroup = TRUE,
  useDemographicsGender = TRUE
)
```

Arguments

```
useDemographicsAgeGroup
    Age of the subject on the index date (in 5 year age groups).
useDemographicsGender
    Gender of the subject.
```

Details

Create simple settings for covariates that can be constructed using only the data in the CaseData object.

Value

An object of type SimpleCovariateSettings

```
fitCaseControlModel    Fit the case-control model
```

Description

Fit the case-control model

Usage

```
fitCaseControlModel(
  caseControlData,
  useCovariates = FALSE,
  excludeCovariateIds = c(),
  includeCovariateIds = c(),
  caseControlsExposure = NULL,
  profileGrid = NULL,
  profileBounds = c(log(0.1), log(10)),
  prior = createPrior("laplace", useCrossValidation = TRUE),
  control = createControl(cvType = "auto", startingVariance = 0.01, tolerance = 2e-07,
    cvRepetitions = 10, selectorType = "byPid", noiseLevel = "quiet")
)
```

Arguments

caseControlData	A data frame as generated by the createCaseControlData function.
useCovariates	Whether to use the covariates in the caseControlsExposure.
excludeCovariateIds	Exclude these covariates from the model.
includeCovariateIds	Include only these covariates in the model.
caseControlsExposure	An object of type caseControlsExposure as created using the getDbExposureData function.
profileGrid	A one-dimensional grid of points on the log(relative risk) scale where the likelihood for coefficient of variables is sampled. See details.
profileBounds	The bounds (on the log relative risk scale) for the adaptive sampling of the likelihood function.
prior	The prior used to fit the model. See createPrior for details.
control	The control object used to control the cross-validation used to determine the hyperparameters of the prior (if applicable). See createControl for details.

Details

Fits the model using a conditional logistic regression.

For likelihood profiling, either specify the 'profileGrid' for a completely user- defined grid, or 'profileBounds' for an adaptive grid. Both should be defined on the log IRR scale. When both 'profileGrid' and 'profileGrid' are 'NULL' likelihood profiling is disabled.

Value

An object of type outcomeModel.

getAttritionTable	<i>Get the attrition table for a population</i>
-------------------	---

Description

Get the attrition table for a population

Usage

```
getAttritionTable(caseControls)
```

Arguments

caseControls	A data frame of cases and controls as generated by the function selectControls .
--------------	--

Value

A data frame specifying the number of cases and events after various steps of filtering.

getDbCaseData	<i>Load case data from the database</i>
---------------	---

Description

Load all data about the cases and nesting cohort from the database.

Usage

```
getDbCaseData(
  connectionDetails,
  cdmDatabaseSchema,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  outcomeDatabaseSchema = cdmDatabaseSchema,
  outcomeTable = "condition_era",
  outcomeIds = c(),
  useNestingCohort = FALSE,
  nestingCohortDatabaseSchema = cdmDatabaseSchema,
  nestingCohortTable = "cohort",
  nestingCohortId = NULL,
  useObservationEndAsNestingEndDate = TRUE,
  getVisits = FALSE,
  getExposures = FALSE,
  exposureDatabaseSchema = cdmDatabaseSchema,
  exposureTable = "drug_era",
  exposureIds = c(),
  studyStartDate = "",
  studyEndDate = "",
  maxNestingCohortSize = 1e+07,
  maxCasesPerOutcome = 5e+05
)
```

Arguments

connectionDetails

An R object of type ConnectionDetails created using the function createConnectionDetails in the DatabaseConnector package.

cdmDatabaseSchema

The name of the database schema that contains the OMOP CDM instance. Requires read permissions to this database. On SQL Server, this should specify both the database and the schema, so for example 'cdm_instance.dbo'.

tempEmulationSchema

A schema where temp tables can be created in Oracle.

outcomeDatabaseSchema

The name of the database schema that is the location where the data used to define the outcome cohorts is available. If outcomeTable = CONDITION_ERA, outcomeDatabaseSchema is not used. Requires read permissions to this database.

outcomeTable

The tablename that contains the outcome cohorts. If outcomeTable is not CONDITION_OCCURRENCE or CONDITION_ERA, then expectation is outcomeTable has format of COHORT table: COHORT_DEFINITION_ID, SUBJECT_ID, COHORT_START_DATE, COHORT_END_DATE.

outcomeIds	A list of ids used to define outcomes. If outcomeTable = CONDITION_OCCURRENCE, the list is a set of ancestor CONCEPT_IDs, and all occurrences of all descendant concepts will be selected. If outcomeTable <> CONDITION_OCCURRENCE, the list contains records found in COHORT_DEFINITION_ID field.
useNestingCohort	Should the study be nested in a cohort (e.g. people with a specific indication)? If not, the study will be nested in the general population.
nestingCohortDatabaseSchema	The name of the database schema that is the location where the nesting cohort is defined.
nestingCohortTable	Name of the table holding the nesting cohort. This table should have the same structure as the cohort table.
nestingCohortId	A cohort definition ID identifying the records in the nestingCohortTable to use as nesting cohort.
useObservationEndAsNestingEndDate	When using a nesting cohort, should the observation period end date be used instead of the cohort end date?
getVisits	Get data on visits? This is needed when matching on visit date is requested later on.
getExposures	Should data on exposures be fetched? All exposure information for the nesting cohort will be retrieved, which may be time-consuming. Usually it is more efficient to fetch exposure data only for the cases and controls, as can be done using the getDbExposureData function.
exposureDatabaseSchema	The name of the database schema that is the location where the exposure data used to define the exposure cohorts is available. If exposureTable = DRUG_ERA, exposureDatabaseSchema is not used but assumed to be cdmSchema. Requires read permissions to this database.
exposureTable	The tablename that contains the exposure cohorts. If exposureTable <> DRUG_ERA, then expectation is exposureTable has format of COHORT table: cohort_concept_id, SUBJECT_ID, COHORT_START_DATE, COHORT_END_DATE.
exposureIds	A list of identifiers to define the exposures of interest. If exposureTable = DRUG_ERA, exposureIds should be CONCEPT_ID. If exposureTable <> DRUG_ERA, exposureIds is used to select the cohort_concept_id in the cohort-like table. If no exposureIds are provided, all drugs or cohorts in the exposureTable are included as exposures.
studyStartDate	A calendar date specifying the minimum date where data is used. Date format is 'yyyymmdd'.
studyEndDate	A calendar date specifying the maximum date where data is used. Date format is 'yyyymmdd'.
maxNestingCohortSize	If the nesting cohort is larger than this number it will be sampled to this size. maxCohortSize = 0 indicates no maximum size.
maxCasesPerOutcome	If there are more than this number of cases for a single outcome cases will be sampled to this size. maxCasesPerOutcome = 0 indicates no maximum size.

Value

Returns an object of type `caseData`, containing information on the cases, the nesting cohort, and optionally visits. Information about multiple outcomes can be captured at once for efficiency reasons. The generic `summary()` function has been implemented for this object.

<code>getDbExposureData</code>	<i>Get exposure data for cases and controls from a database</i>
--------------------------------	---

Description

If additional exposure data or covariate data is required, this function will send the information on the cases and controls back to the server.

Note: For PDW and RedShift, where uploading data can be excruciatingly slow, you can use bulk loading by preparing the environment as described in the [insertTable](#) function in the DatabaseConnection package, and setting `tempEmulationSchema` to a schema where you have write privileges (bulk import can only upload to permanent tables).

Usage

```
getDbExposureData(
  caseControls,
  connectionDetails,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  exposureDatabaseSchema = NULL,
  exposureTable = "drug_era",
  exposureIds = c(),
  cdmDatabaseSchema = exposureDatabaseSchema,
  covariateSettings = NULL,
  caseData = NULL
)
```

Arguments

`caseControls` A data frame as generated by the [selectControls](#) function.

`connectionDetails`

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

`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.

`exposureDatabaseSchema`

The name of the database schema that is the location where the exposure data used to define the exposure cohorts is available. If `exposureTable = DRUG_ERA`, `exposureDatabaseSchema` is not used but assumed to be `cdmSchema`. Requires read permissions to this database.

`exposureTable`

The tablename that contains the exposure cohorts. If `exposureTable <> drug_era`, then expectation is `exposureTable` has format of COHORT table: `cohort_definition_id, subject_id, cohort_start_date, cohort_end_date`.

exposureIds	A list of identifiers to define the exposures of interest. If exposureTable = drug_era, exposureIds should be concept_id. If exposureTable <> drug_era, exposureIds is used to select the cohort_definition_id in the cohort-like table. If no exposureIds are provided, all drugs or cohorts in the exposureTable are included as exposures.
cdmDatabaseSchema	Needed when constructing covariates using FeatureExtraction: the name of the database schema that contains the OMOP CDM instance. Requires read permissions to this database. On SQL Server, this should specify both the database and the schema, so for example 'cdm_instance.dbo'.
covariateSettings	Either an object of type covariateSettings as created using the createCovariateSettings function in the FeatureExtraction package, or an object of type SimpleCovariateSettings as created using the createSimpleCovariateSettings function. If NULL then no covariate data is retrieved.
caseData	An object of type caseData as generated using the getDbCaseData function. If caseData is provided and contains the exposure data (see getExposures in the getDbCaseData function, and if no covariates need to be constructed (covariateSettings = NULL) or only simple covariates need to be constructed, then the no connection to the database is used to create the exposure data. This may be much more efficient in some situations.

insertDbPopulation *Insert cases and controls into a database*

Description

Insert cases and controls into a database

Usage

```
insertDbPopulation(
  caseControls,
  cohortIds = c(1, 0),
  connectionDetails,
  cohortDatabaseSchema,
  cohortTable = "cohort",
  createTable = FALSE,
  dropTableIfExists = TRUE
)
```

Arguments

caseControls	A data frame as generated by the selectControls function.
cohortIds	The IDs to be used for the cohorts of cases and controls, respectively.
connectionDetails	An R object of type connectionDetails created using the function createConnectionDetails in the DatabaseConnector package.

cohortDatabaseSchema	The name of the database schema where the data will be written. Requires write permissions to this database. On SQL Server, this should specify both the database and the schema, so for example 'cdm_instance.dbo'.
cohortTable	The name of the table in the database schema where the data will be written.
createTable	Should a new table be created? If not, the data will be inserted into an existing table.
dropTableIfExists	If createTable = TRUE and the table already exists it will be overwritten.

Details

Inserts cases and controls into a database. The table in the database will have the same structure as the 'cohort' table in the Common Data Model.

isCaseData	<i>Check whether an object is a CaseData object</i>
------------	---

Description

Check whether an object is a CaseData object

Usage

isCaseData(x)

Arguments

x The object to check.

Value

A logical value.

loadCaseControlsExposure	<i>Load the caseControlsExposure data from a folder</i>
--------------------------	---

Description

loadCaseControlsExposure loads an object of type caseControlsExposure from a folder in the file system.

Usage

loadCaseControlsExposure(folder, readOnly = TRUE)

Arguments

folder The name of the folder containing the data.
readOnly If true, the data is opened read only.

Details

The data will be written to a set of files in the folder specified by the user.

Value

An object of class caseControlsExposure.

loadCaseData	<i>Load the covariate data from a folder</i>
--------------	--

Description

loadCaseData loads an object of type caseData from a file in the file system.

Usage

```
loadCaseData(file)
```

Arguments

file The name of the file containing the data.

Details

The data will be read from the file specified by the user.

Value

An object of class CaseData.

loadCcAnalysisList	<i>Load a list of ccAnalysis from file</i>
--------------------	--

Description

Load a list of objects of type ccAnalysis from file. The file is in JSON format.

Usage

```
loadCcAnalysisList(file)
```

Arguments

file The name of the file

Value

A list of objects of type ccAnalysis.

loadExposureOutcomeNestingCohortList

Load a list of exposureOutcomeNestingCohort from file

Description

Load a list of objects of type exposureOutcomeNestingCohort from file. The file is in JSON format.

Usage

```
loadExposureOutcomeNestingCohortList(file)
```

Arguments

file The name of the file

Value

A list of objects of type drugComparatorOutcome.

runCcAnalyses

Run a list of analyses

Description

Run a list of analyses

Usage

```
runCcAnalyses(
  connectionDetails,
  cdmDatabaseSchema,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  exposureDatabaseSchema = cdmDatabaseSchema,
  exposureTable = "drug_era",
  outcomeDatabaseSchema = cdmDatabaseSchema,
  outcomeTable = "condition_era",
  nestingCohortDatabaseSchema = cdmDatabaseSchema,
  nestingCohortTable = "condition_era",
  outputFolder = "./CcOutput",
  ccAnalysisList,
  exposureOutcomeNestingCohortList,
  prefetchExposureData = FALSE,
  getDbCaseDataThreads = 1,
  selectControlsThreads = 1,
  getDbExposureDataThreads = 1,
  createCaseControlDataThreads = 1,
  fitCaseControlModelThreads = 1,
  cvThreads = 1
)
```

Arguments

- connectionDetails**
An R object of type `ConnectionDetails` created using the function `createConnectionDetails` in the `DatabaseConnector` package.
- cdmDatabaseSchema**
The name of the database schema that contains the OMOP CDM instance. Requires read permissions to this database. On SQL Server, this should specify both the database and the schema, so for example `'cdm_instance.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.
- exposureDatabaseSchema**
The name of the database schema that is the location where the exposure data used to define the exposure cohorts is available. If `exposureTable = DRUG_ERA`, `exposureDatabaseSchema` is not used but assumed to be `cdmSchema`. Requires read permissions to this database.
- exposureTable**
The tablename that contains the exposure cohorts. If `exposureTable <> drug_era`, then expectation is `exposureTable` has format of COHORT table: `cohort_definition_id, subject_id, cohort_start_date, cohort_end_date`.
- outcomeDatabaseSchema**
The name of the database schema that is the location where the data used to define the outcome cohorts is available. If `outcomeTable = CONDITION_ERA`, `outcomeDatabaseSchema` is not used. Requires read permissions to this database.
- outcomeTable**
The tablename that contains the outcome cohorts. If `outcomeTable` is not `CONDITION_OCCURRENCE` or `CONDITION_ERA`, then expectation is `outcomeTable` has format of COHORT table: `COHORT_DEFINITION_ID, SUBJECT_ID, COHORT_START_DATE, COHORT_END_DATE`.
- nestingCohortDatabaseSchema**
The name of the database schema that is the location where the nesting cohort is defined.
- nestingCohortTable**
Name of the table holding the nesting cohort. This table should have the same structure as the cohort table.
- outputFolder**
Name of the folder where all the outputs will be written to.
- ccAnalysisList**
A list of objects of type `ccAnalysis` as created using the `createCcAnalysis` function.
- exposureOutcomeNestingCohortList**
A list of objects of type `exposureOutcomeNestingCohort` as created using the `createExposureOutcomeNestingCohort` function.
- prefetchExposureData**
Should exposure data for the entire nesting cohort be fetched at the beginning, or should exposure data be fetched later specifically for a set of cases and controls. Prefetching can be faster when there are many outcomes but only few exposures. Prefetching does not speed up performance when covariates also need to be constructed.
- getDbCaseDataThreads**
The number of parallel threads to use for building the `caseData` objects.
- selectControlsThreads**
The number of parallel threads to use for selecting controls.

getDbExposureDataThreads	The number of parallel threads to use for fetching data on exposures for cases and controls.
createCaseControlDataThreads	The number of parallel threads to use for creating case and control data including exposure status indicators
fitCaseControlModelThreads	The number of parallel threads to use for fitting the models.
cvThreads	The number of parallel threads used for the cross-validation to determine the hyper-parameter when fitting the model.

Details

Run a list of analyses for the exposure-outcome-nesting cohorts of interest. This function will run all specified analyses against all hypotheses of interest, meaning that the total number of outcome models is `length(ccAnalysisList) * length(exposureOutcomeNestingCohortList)` (if all analyses specify an outcome model should be fitted). When you provide several analyses it will determine whether any of the analyses have anything in common, and will take advantage of this fact. For example, if we specify several analyses that only differ in the way the outcome model is fitted, then this function will extract the data and fit the propensity model only once, and re-use this in all the analysis.

saveCaseControlsExposure

Save the caseControlsExposure data to folder

Description

saveCaseControlsExposure saves an object of type caseControlsExposure to folder.

Usage

```
saveCaseControlsExposure(caseControlsExposure, folder)
```

Arguments

caseControlsExposure	An object of type caseControlsExposure as generated using getDbExposureData .
folder	The name of the folder where the data will be written. The folder should not yet exist.

Details

The data will be written to a set of files in the specified folder.

saveCaseData	<i>Save the case data to file</i>
--------------	-----------------------------------

Description

saveCaseData saves an object of type caseData to file

Usage

```
saveCaseData(caseData, file)
```

Arguments

caseData	An object of type CaseData as generated using getDbCaseData .
file	The name of the file where the data will be written. If the file exists it will be overwritten.

Details

The data will be written to the file specified by the user.

saveCcAnalysisList	<i>Save a list of ccAnalysis to file</i>
--------------------	--

Description

Write a list of objects of type ccAnalysis to file. The file is in JSON format.

Usage

```
saveCcAnalysisList(ccAnalysisList, file)
```

Arguments

ccAnalysisList	The ccAnalysis list to be written to file
file	The name of the file where the results will be written

```
saveExposureOutcomeNestingCohortList
    Save a list of drugComparatorOutcome to file
```

Description

Write a list of objects of type exposureOutcomeNestingCohort to file. The file is in JSON format.

Usage

```
saveExposureOutcomeNestingCohortList(exposureOutcomeNestingCohortList, file)
```

Arguments

```
exposureOutcomeNestingCohortList
    The exposureOutcomeNestingCohort list to be written to file
file
    The name of the file where the results will be written
```

```
selectControls    Select controls
```

Description

Select controls

Usage

```
selectControls(
  caseData,
  outcomeId,
  firstOutcomeOnly = TRUE,
  washoutPeriod = 180,
  controlSelectionCriteria = createMatchingCriteria(),
  minAge = NULL,
  maxAge = NULL
)
```

Arguments

```
caseData      An object of type caseData as generated using the getDbCaseData function.
outcomeId     The outcome ID of the cases for which we need to pick controls.
firstOutcomeOnly
    Use the first outcome per person?
washoutPeriod Minimum required numbers of days of observation for inclusion as either case
    or control.
controlSelectionCriteria
    Either a matchingCriteria object as generated using the createMatchingCriteria
    function, or a samplingCriteria object as generated using the createSamplingCriteria
    function.
```


minAge	Minimum age at which patient time will be included in the analysis. Note that information prior to the min age is still used to determine exposure status after the minimum age (e.g. when a prescription was started just prior to reaching the minimum age). Also, outcomes occurring before the minimum age is reached will be considered as prior outcomes when using first outcomes only. Age should be specified in years, but non-integer values are allowed. If not specified, no age restriction will be applied.
maxAge	Maximum age at which patient time will be included in the analysis. Age should be specified in years, but non-integer values are allowed. If not specified, no age restriction will be applied.

Details

Select controls either by individually matching controls to each case, or by random sampling controls from the (nested) population.

Value

A data frame with these columns:

personId The person ID

indexDate The index date

isCase Is the person a case or a control?

stratumId The ID linking cases and controls in a matched set (only available when matching)

summarizeCcAnalyses *Create a summary report of the analyses*

Description

Create a summary report of the analyses

Usage

```
summarizeCcAnalyses(outcomeReference, outputFolder)
```

Arguments

outcomeReference

A data.frame as created by the [runCcAnalyses](#) function.

outputFolder

Name of the folder where all the outputs have been written to.

Index

Andromeda, [2](#)

CaseData (CaseData-class), [2](#)
CaseData-class, [2](#)
computeMdr, [3](#)
createCaseControlData, [3](#), [4](#), [13](#)
createCcAnalysis, [5](#), [7](#), [21](#)
createControl, [13](#)
createCreateCaseControlDataArgs, [5](#), [6](#)
createExposureOutcomeNestingCohort, [6](#),
[21](#)
createFitCaseControlModelArgs, [5](#), [7](#)
createGetDbCaseDataArgs, [5](#), [8](#)
createGetDbExposureDataArgs, [5](#), [9](#)
createMatchingCriteria, [9](#), [24](#)
createPrior, [13](#)
createSamplingCriteria, [10](#), [24](#)
createSelectControlsArgs, [5](#), [11](#)
createSimpleCovariateSettings, [12](#), [17](#)

fitCaseControlModel, [12](#)

getAttritionTable, [13](#)
getDbCaseData, [2](#), [14](#), [17](#), [23](#), [24](#)
getDbExposureData, [4](#), [13](#), [15](#), [16](#), [22](#)

insertDbPopulation, [17](#)
insertTable, [16](#)
isCaseData, [3](#), [18](#)

loadCaseControlsExposure, [18](#)
loadCaseData, [2](#), [19](#)
loadCcAnalysisList, [19](#)
loadExposureOutcomeNestingCohortList,
[20](#)

runCcAnalyses, [6](#), [7](#), [20](#), [25](#)

saveCaseControlsExposure, [22](#)
saveCaseData, [2](#), [23](#)
saveCcAnalysisList, [23](#)
saveExposureOutcomeNestingCohortList,
[24](#)

selectControls, [10](#), [13](#), [16](#), [17](#), [24](#)
show, CaseData-method (CaseData-class), [2](#)
summarizeCcAnalyses, [25](#)
summary, CaseData-method
(CaseData-class), [2](#)
Sys.time, [10](#)