

Package ‘CaseControl’

November 14, 2017

Type Package

Title Case-Control

Version 1.4.0

Date 2017-11-14

Author Martijn Schuemie

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

Depends R (>= 3.2.2),
Cyclops (>= 1.3.0),
DatabaseConnector (>= 2.0.0),
survival,
FeatureExtraction (>= 2.0.0)

Imports SqlRender (>= 1.4.5),
bit,
ff,
ffbase (>= 0.12.1),
Rcpp (>= 0.11.2),
OhdsiRTools (>= 1.1.1),
plyr

Suggests testthat,
knitr,
rmarkdown,
EmpiricalCalibration

License Apache License 2.0

LinkingTo Rcpp

NeedsCompilation yes

RoxygenNote 6.0.1

R topics documented:

CaseControl	2
computeMdr	2
createCaseControlData	3

createCcAnalysis	4
createCreateCaseControlDataArgs	5
createExposureOutcomeNestingCohort	5
createFitCaseControlModelArgs	6
createGetDbCaseDataArgs	7
createGetDbExposureDataArgs	7
createSelectControlsArgs	8
fitCaseControlModel	9
getAttritionTable	10
getDbCaseData	10
getDbExposureData	12
insertDbPopulation	13
loadCaseControlsExposure	14
loadCaseData	14
loadCcAnalysisList	15
loadExposureOutcomeNestingCohortList	15
runCcAnalyses	16
saveCaseControlsExposure	18
saveCaseData	18
saveCcAnalysisList	19
saveExposureOutcomeNestingCohortList	19
selectControls	19
summarizeCcAnalyses	21

Index 22

CaseControl	<i>CaseControl</i>
Description	
CaseControl	
computeMdr	<i>Compute the minimum detectable relative risk</i>

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)
```

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. |

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	Create a case-control analysis specification
------------------	--

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)
```

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.

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 indentifying 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 indentifying 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 indentifying 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(),
  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.
prior	The prior used to fit the model. SeecreatePrior for details.
control	The control object used to control the cross-validation used todetermine the hyperparameters of the prior (if applicable). SeecreateControl 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 = TRUE,
  studyStartDate = "", studyEndDate = "")
```

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'.

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

An object of type covariateSettings as created using the createCovariateSettings function in the FeatureExtraction package. If NULL then no covariate data is retrieved.

Details

Create an object defining the parameter values.

createSelectControlsArgs

Create a parameter object for the function selectControls

Description

Create a parameter object for the function selectControls

Usage

```
createSelectControlsArgs(firstOutcomeOnly = TRUE, washoutPeriod = 180,
  controlsPerCase = 2, matchOnAge = TRUE, ageCaliper = 2,
  matchOnGender = TRUE, matchOnProvider = FALSE, matchOnCareSite = FALSE,
  matchOnVisitDate = FALSE, visitDateCaliper = 30,
  matchOnTimeInCohort = FALSE, daysInCohortCaliper = 30, minAge = NULL,
  maxAge = NULL, removedUnmatchedCases = TRUE)
```

Arguments

firstOutcomeOnly

Use the first outcome per person?

washoutPeriod Minimum required numbers of days of observation for inclusion as either case or control.

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 visitdate?

visitDateCaliper

Maximum difference (in days) between the index date and the visitdate 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.
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.
removedUnmatchedCases	Should cases with no matched controls be removed?

Details

Create an object defining the parameter values.

fitCaseControlModel	<i>Fit the case-control model</i>
---------------------	-----------------------------------

Description

Fit the case-control model

Usage

```
fitCaseControlModel(caseControlData, useCovariates = FALSE,
  excludeCovariateIds = c(), includeCovariateIds = c(),
  caseControlsExposure = NULL, 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.

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.

Value

An object of type `outcomeModel`.

<code>getAttritionTable</code>	<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.

<code>getDbCaseData</code>	<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,
  oracleTempSchema = cdmDatabaseSchema,
  outcomeDatabaseSchema = cdmDatabaseSchema, outcomeTable = "condition_era",
  outcomeIds = c(), useNestingCohort = FALSE,
  nestingCohortDatabaseSchema = cdmDatabaseSchema,
  nestingCohortTable = "cohort", nestingCohortId = NULL,
  useObservationEndAsNestingEndDate = TRUE, getVisits = TRUE,
  getExposures = FALSE, exposureDatabaseSchema = cdmDatabaseSchema,
  exposureTable = "drug_era", exposureIds = c(), studyStartDate = "",
  studyEndDate = "")
```

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'.
oracleTempSchema	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'.

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.

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

Description

Get exposure data for cases and controls from a database

Usage

```
getDbExposureData(caseControls, connectionDetails, oracleTempSchema = NULL,
  exposureDatabaseSchema, 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.
oracleTempSchema	A schema where temp tables can be created in Oracle.
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: 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	An object of type covariateSettings as created using the createCovariateSettings function in the FeatureExtraction package. 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), then the no connection to the database is used to create the exposure data. This may be much more efficient in some situations.

insertDbPopulation	<i>Insert cases and controls into a database</i>
--------------------	--

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.

loadCaseControlsExposure

Load the caseControlsExposure data from a folder

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

Load the case data from a folder

Description

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

Usage

```
loadCaseData(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 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	<i>Load a list of exposureOutcomeNestingCohort from file</i>
--------------------------------------	--

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,
  oracleTempSchema = cdmDatabaseSchema,
  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'.

oracleTempSchema

A schema where temp tables can be created in Oracle.

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 analyses.

`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`*Save the case data to folder*

Description

`saveCaseData` saves an object of type `caseData` to folder.

Usage

```
saveCaseData(caseData, folder)
```

Arguments

`caseData`

An object of type `caseData` as generated using [getDbCaseData](#).

`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.

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	<i>Save a list of drugComparatorOutcome to file</i>
--------------------------------------	---

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	<i>Select matched controls per case</i>
----------------	---

Description

Select matched controls per case

Usage

```
selectControls(caseData, outcomeId, firstOutcomeOnly = TRUE,
  washoutPeriod = 180, controlsPerCase = 2, matchOnAge = TRUE,
  ageCaliper = 2, matchOnGender = TRUE, matchOnProvider = FALSE,
  matchOnCareSite = FALSE, matchOnVisitDate = FALSE,
  visitDateCaliper = 30, matchOnTimeInCohort = FALSE,
  daysInCohortCaliper = 30, minAge = NULL, maxAge = NULL,
  removedUnmatchedCases = TRUE)
```

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.
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.
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.
removedUnmatchedCases	Should cases with no matched controls be removed?

Details

Select controls per case. Controls are matched on calendar time and the criteria defined in the arguments. Controls are randomly sampled to the required number.

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

summarizeCcAnalyses	<i>Create a summary report of the analyses</i>
---------------------	--

Description

Create a summary report of the analyses

Usage

```
summarizeCcAnalyses(outcomeReference)
```

Arguments

outcomeReference

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

Index

CaseControl, [2](#)
CaseControl-package (CaseControl), [2](#)
computeMdr, [2](#)
createCaseControlData, [2](#), [3](#), [9](#)
createCcAnalysis, [4](#), [6](#), [17](#)
createControl, [10](#)
createCreateCaseControlDataArgs, [4](#), [5](#)
createExposureOutcomeNestingCohort, [5](#),
[17](#)
createFitCaseControlModelArgs, [4](#), [6](#)
createGetDbCaseDataArgs, [4](#), [7](#)
createGetDbExposureDataArgs, [4](#), [7](#)
createPrior, [10](#)
createSelectControlsArgs, [4](#), [8](#)

fitCaseControlModel, [9](#)

getAttritionTable, [10](#)
getDbCaseData, [10](#), [13](#), [18](#), [20](#)
getDbExposureData, [3](#), [9](#), [11](#), [12](#), [18](#)

insertDbPopulation, [13](#)

loadCaseControlsExposure, [14](#)
loadCaseData, [14](#)
loadCcAnalysisList, [15](#)
loadExposureOutcomeNestingCohortList,
[15](#)

runCcAnalyses, [5](#), [6](#), [16](#), [21](#)

saveCaseControlsExposure, [18](#)
saveCaseData, [18](#)
saveCcAnalysisList, [19](#)
saveExposureOutcomeNestingCohortList,
[19](#)
selectControls, [10](#), [12](#), [13](#), [19](#)
summarizeCcAnalyses, [21](#)