Package 'IcTemporalPatternDiscovery'

July 5, 2017

Type Package

Title IC Temporal Pattern Discovery

Version 1.0.2

Date 2017-07-05

Author Tomas Bergvall

Maintainer Tomas Bergvall <tomas.bergvall@who-umc.org>

Description Population-level estimation method that estimates risk by combining a self-controlled and cohort design.

Depends DatabaseConnector (>= 1.11.4),

Imports SqlRender, OhdsiRTools (>= 1.1.1)

License Apache License 2.0

RoxygenNote 6.0.1

R topics documented:

`calculateStatisticsIc` *compute the IC statistics*

Description

Computes the IC statistics.

Usage

```
calculateStatisticsIc(ictpdData, multipleControlPeriods = "110",
  multipleRiskPeriods = "10000", shrinkage = 0.5, icPercentile = 0.025,
  metric = "IC025")
```

Arguments

`ictpdData`	An object containing the counts, as created using the `getDbIctpdData` function.
`multipleControlPeriods`	Defines the control periods to use where 100 means the control period defined by controlPeriodStart/End, 010 means the period -30 to -1 day before prescription and 001 means the control period on the day of prescription
`multipleRiskPeriods`	Defines the risk periods to use 10000 is 1-30 days, 01000 is 1 to 360 days, 00100 is 31 to 90 days, 00010 is 91 to 180 and 00001 is 721 to 1080 days after prescription default is '10000'
`shrinkage`	Shrinkage used in IRR calculations, required >0 to deal with 0 case counts, but larger number means more shrinkage. default is 0.5
`icPercentile`	The lower bound of the credibility interval for the IC values (IClow). default is 0.025,
`metric`	Defines wether the output will contain the point estimate or the lower bound. Available input is 'IC and 'IC025' default is 'IC025'

Value

An object of type `ictpdResults` containing the results.

Examples

```
## Not run:
library(SelfControlledCohort)

connectionDetails <- createConnectionDetails(dbms = "postgresql",
                                             user = "joe",
                                             password = "secret",
                                             server = "myserver")
exposureOutcomePairs <- data.frame(outcomeId = c(196794, 196794, 312648),
                                   exposurId = c(1501700, 1545958, 1551803))
ictpdData <- getDbIctpdData(connectionDetails,
                            cdmDatabaseSchema = "cdm_schema.dbo",
                            exposureOutcomePairs = exposureOutcomePairs)
ictpdResults <- calculateStatisticsIC(ictpdData)
ictpdResults
```

```
## End(Not run)
```

---

createCalculateStatisticsIcArgs *Create a parameter object for the function calculateStatisticsIc*

Description

Create a parameter object for the function calculateStatisticsIc

Usage

```
createCalculateStatisticsIcArgs(multipleControlPeriods = "110",
  multipleRiskPeriods = "10000", shrinkage = 0.5, icPercentile = 0.025,
  metric = "IC025")
```

Arguments

multipleControlPeriods	Defines the control periods to use where 100 means the controlperiod defined by controlPeriodStart/End, 010 means the period -30to -1 day before prescription and 001 means the control period onthe day of prescription
multipleRiskPeriods	Defines the risk periods to use 10000 is 1-30 days, 01000 is 1 to360 days, 00100 is 31 to 90 days, 00010 is 91 to 180 and 00001 is721 to 1080 days after prescription default is '10000'
shrinkage	Shrinkage used in IRR calculations, required >0 to deal with 0 casecounts, but larger number means more shrinkage. default is 0.5
icPercentile	The lower bound of the credibility interval for the IC values(IClow). default is 0.025,
metric	Defines wether the output will contain the point estimate or thelower bound. Available input is 'IC and 'IC025' default is 'IC025'

Details

Create an object defining the parameter values.

---

createExposureOutcome *Create exposure-outcome combinations.*

Description

Create exposure-outcome combinations.

Usage

```
createExposureOutcome(exposureId, outcomeId)
```

Arguments

exposureId	A concept ID indentifying the drug of interest 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 createIctpdAnalysis function.
outcomeId	A concept ID indentifying the outcome of interest 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 createIctpdAnalysis function.

Details

Create a hypothesis of interest, to be used with the runIctpdAnalyses function.

---

createGetDbIctpdDataArgs

Create a parameter object for the function getDbIctpdData

Description

Create a parameter object for the function getDbIctpdData

Usage

```
createGetDbIctpdDataArgs(drugTypeConceptIdList = c(38000182),
  conditionTypeConceptIdList = c(38000247), controlPeriodStart = -1080,
  controlPeriodEnd = -361, riskPeriodStart = 1, riskPeriodEnd = 30,
  censor = FALSE)
```

Arguments

drugTypeConceptIdList	Which drug_type to use: generally only use 1 value (ex: 30dera).
conditionTypeConceptIdList	Which condition_type to use: generally only use 1 value (ex: 30dera).
controlPeriodStart	start of the control period - can be set between -99999 and0, default is -1080.
controlPeriodEnd	end of the control period - can be set between -99999 and0, default is -361.
riskPeriodStart	start of the risk period - can be set between 0 and 99999,default is 1.
riskPeriodEnd	end of the risk period - can be set between 0 and 99999,default is 30.
censor	a flag indicating wether the method should censor theobservation period at the end of exposure or not. Availableinput is 0 or 1 with default = 0.

Details

Create an object defining the parameter values.

createIctpdAnalysis *Create ICTPD analysis details*

Description

createIctpdAnalysis generates an object specifying one set of analysis choices for the IC Temporal Pattern Discovery method.

Usage

```
createIctpdAnalysis(analysisId = 1, description = "", exposureType = NULL,
  outcomeType = NULL, getDbIctpdDataArgs, calculateStatisticsIcArgs)
```

Arguments

analysisId	A unique identifier that can later be used to identify the results of this analysis
description	A short description of the analysis.
exposureType	If more than one exposure is provided for each exposureOutcome, 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 exposureOutcome, this field should be used to select the specific outcome to use in this analysis.
getDbIctpdDataArgs	An object representing the arguments to be used when calling the getDbIctpdData function.
calculateStatisticsIcArgs	An object representing the arguments to be used when calling the calculateStatisticsIc function.

getDbIctpdData *Get ICTPD counts from database*

Description

This function is used to load the counts needed to compute the ICTPD from a database in OMOP CDM format.

Usage

```
getDbIctpdData(connectionDetails, cdmDatabaseSchema,
  oracleTempSchema = cdmDatabaseSchema, cdmVersion = "4",
  exposureOutcomePairs, exposureDatabaseSchema = cdmDatabaseSchema,
  exposureTable = "drug_era", outcomeDatabaseSchema = cdmDatabaseSchema,
  outcomeTable = "condition_era", drugTypeConceptIdList = c(38000182),
  conditionTypeConceptIdList = c(38000247), controlPeriodStart = -1080,
  controlPeriodEnd = -361, riskPeriodStart = 1, riskPeriodEnd = 30,
  censor = FALSE)
```

Arguments

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

`cdmDatabaseSchema`
: Name of database schema that contains OMOP CDM and vocabulary.

`oracleTempSchema`
: For Oracle only: the name of the database schema where you want all temporary tables to be managed. Requires create/insert permissions to this database.

`cdmVersion`
: Define the OMOP CDM version used: currently support "4" and "5".

`exposureOutcomePairs`
: A data frame with at least two columns:
  - "exposureId" containing the drug_concept_ID or cohort_concept_id of the exposure variable
  - "outcomeId" containing the condition_concept_ID or cohort_concept_id of the outcome variable

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

`outcomeTable`
: The tablename that contains the outcome cohorts. If outcomeTable <> CONDITION_OCCURRENCE, then expectation is outcomeTable has format of COHORT table: COHORT_DEFINITION_ID, SUBJECT_ID, COHORT_START_DATE, COHORT_END_DATE.

`drugTypeConceptIdList`
: Which drug_type to use: generally only use 1 value (ex: 30d era).

`conditionTypeConceptIdList`
: Which condition_type to use: generally only use 1 value (ex: 30d era).

`controlPeriodStart`
: start of the control period - can be set between -99999 and 0, default is -1080.

`controlPeriodEnd`
: end of the control period - can be set between -99999 and 0, default is -361.

`riskPeriodStart`
: start of the risk period - can be set between 0 and 99999, default is 1.

`riskPeriodEnd`
: end of the risk period - can be set between 0 and 99999, default is 30.

`censor`
: a flag indicating wether the method should censor the observation period at the end of exposure or not. Available input is 0 or 1 with default = 0.

Value

An object of type `ictpdData` containing counts that can be used in the `calculateStatisticsIc` function.

Examples

```
## Not run:
library(SelfControlledCohort)

connectionDetails <- createConnectionDetails(dbms = "postgresql",
                                             user = "joe",
                                             password = "secret",
                                             server = "myserver")
exposureOutcomePairs <- data.frame(outcomeId = c(196794, 196794, 312648),
                                   exposurId = c(1501700, 1545958, 1551803))
ictpdData <- getDbIctpdData(connectionDetails,
                            cdmDatabaseSchema = "cdm_schema.dbo",
                            exposureOutcomePairs = exposureOutcomePairs)
ictpdResults <- calculateStatisticsIC(ictpdData)
ictpdResults

## End(Not run)
```

`ICTemporalPatternDiscovery` *ICTemporalPatternDiscovery*

Description

ICTemporalPatternDiscovery

`loadExposureOutcomeList` *Load a list of exposureOutcome from file*

Description

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

Usage

```
loadExposureOutcomeList(file)
```

Arguments

`file`	The name of the file

Value

A list of objects of type `exposureOutcome`.

loadIctpdAnalysisList *Load a list of ictpdAnalysis from file*

Description

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

Usage

```
loadIctpdAnalysisList(file)
```

Arguments

`file`	The name of the file

Value

A list of objects of type `ictpdAnalysis`.

runIctpdAnalyses *Run a list of analyses*

Description

Run a list of analyses

Usage

```
runIctpdAnalyses(connectionDetails, cdmDatabaseSchema,
  oracleTempSchema = cdmDatabaseSchema,
  exposureDatabaseSchema = cdmDatabaseSchema, exposureTable = "drug_era",
  outcomeDatabaseSchema = cdmDatabaseSchema, outcomeTable = "condition_era",
  cdmVersion = 4, outputFolder = "./IctpdOutput", ictpdAnalysisList,
  exposureOutcomeList, getDbIctpdDataThreads = 1,
  calculateStatisticsIcThreads = 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 specifiy both the database and the schema, so for example 'cdm_instance.dbo'.

`oracleTempSchema`
: For Oracle only: the name of the database schema where you want all temporary tables to be managed. Requires create/insert permissions to this database.

`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 by 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 exposureTable = CONDITION_ERA, exposureDatabaseSchema is not used by assumed to be cdmSchema. Requires read permissions to this database.

`outcomeTable`
: The tablename that contains the outcome cohorts. If outcomeTable <> CONDITION_OCCURRENCE, then expectation is outcomeTable has format of COHORT table: COHORT_DEFINITION_ID, SUBJECT_ID, COHORT_START_DATE, COHORT_END_DATE.

`cdmVersion`
: Define the OMOP CDM version used: currently support "4" and "5".

`outputFolder`
: Name of the folder where all the outputs will written to.

`ictpdAnalysisList`
: A list of objects of type `ictpdAnalysis` as created using the `createIctpdAnalysis` function.

`exposureOutcomeList`
: A list of objects of type `exposureOutcome` as created using the `createExposureOutcome` function.

`getDbIctpdDataThreads`
: The number of parallel threads to use to load the data from the database.

`calculateStatisticsIcThreads`
: The number of threads used to perform the IC statistics computations.

Details

Run a list of analyses for the exposure-outcomes of interest. This function will run all specified analyses against all hypotheses of interest, meaning that the total number of outcome models is 'length(ictpdAnalysisList) * length(exposureOutcomeList)'. 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.

`saveExposureOutcomeList` *Save a list of exposureOutcome to file*

Description

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

Usage

```
saveExposureOutcomeList(exposureOutcomeList, file)
```

Arguments

exposureOutcomeList	The exposureOutcome list to be written to file
file	The name of the file where the results will be written

---

saveIctpdAnalysisList *Save a list of ictpdAnalysis to file*

Description

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

Usage

```
saveIctpdAnalysisList(ictpdAnalysisList, file)
```

Arguments

ictpdAnalysisList	The ictpdAnalysis list to be written to file
file	The name of the file where the results will be written

---

summarizeAnalyses *Create a summary report of the analyses*

Description

Create a summary report of the analyses

Usage

```
summarizeAnalyses(resultsReference)
```

Arguments

resultsReference	A data.frame as created by the runIctpdAnalyses function.

Index