Package 'IcTemporalPatternDiscovery'

November 14, 2019

Type Package

Title IC Temporal Pattern Discovery

Version 1.1.1

Date 2018-10-09

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

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

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

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

Depends DatabaseConnector (>= 1.11.4),

Imports SqlRender,
ParallelLogger,
ggplot2,
gridExtra

License Apache License 2.0

RoxygenNote 7.0.0

Encoding UTF-8

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, e.g. a list with names and values corresponding to a call to `getDbIctpdData`. It's convenient to use `createGetDbIctpdDataArgs` on the returning object of `getDbIctpdData`.

`calculateStatisticsIcArgs`
An object representing the arguments to be used when calling the `calculateStatisticsIc` function, analogous to getDbIctpdDataArgs.

Value

Returns a list, with type set to "ictpdAnalysis", with the input parameters as list elements.

Examples

```
# Example 1. "Use the args-functions on a prespecificed analysis"

# Set up (for use with synpuf5pct-data)
connectionDetails <- createConnectionDetails(dbms ="sql server", server="UMCDB06", schema="OmopCdm")
conn <- DatabaseConnector::connect(connectionDetails)
cdmDatabaseSchema <- "synpuf5pct_20180710"
exposureIds = c(40170549, 1321636, 40167333, 711452, 40173533, 981774, 19030860, 19037401, 925102, 1337068)
outcomeIds = c(321389, 73609, 134401, 74725, 375251, 199063, 381580, 4046804, 442094, 436883)
exposureOutcomePairs_example = data.frame("exposureId"=exposureIds, "outcomeId"=outcomeIds)

# Create an analysis
prespec_analysis <- getChronographData(connectionDetails = connectionDetails,

# Store the input parameters of the analysis
data_params <- createGetDbIctpdDataArgs(prespec_analysis)

createIctpdAnalysis(analysisId = 1, "An example of input parameters", getDbIctpdDataArgs=data_params)
```

getChronographData *Get the data for chronographs from the server.*

Description

Get the data for creating chronographs from the server.

Usage

```
getChronographData(
  connectionDetails,
  cdmDatabaseSchema,
  oracleTempSchema = NULL,
  cdmVersion = "5",
  exposureIds = c(),
  outcomeIds = c(),
  exposureOutcomePairs = NULL,
  exposureDatabaseSchema = cdmDatabaseSchema,
  exposureTable = "drug_era",
  outcomeDatabaseSchema = cdmDatabaseSchema,
  outcomeTable = "condition_era",
  shrinkage = 0.5,
  icPercentile = 0.025,
  returnSQL = F
)
```

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 supports "5" and "6".

`exposureIds`
: A vector of IDs identifying the exposures to include when computing the expected count, i.e. the supplied IDs will define the background population in the chronograph. If the exposure table is the drug_era table, these IDs correspond to ingredient concept IDs. If the exposure table has the format of the cohort table, these IDs correspond to the cohort definition IDs. If left empty, all records in the exposure table will be used.

`outcomeIds`
: A vector of IDs identifying the outcomes to include when computing the expected count, i.e. the supplied IDs will define the background population in the chronograph. If the outcome table is the drug_era table, these IDs correspond to condition concept IDs. If the outcomes table has the format of the cohort table, these IDs correspond to the cohort definition IDs. If left empty, all records in the outcome table will be used.

`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

Each row specifies an exposure-outcome-combination for which to collect data for. If left empty, all possible combinations of exposures and outcomes will be computed, which could be time consuming. To collect data for plotting multiple exposures and outcomes in a single chronograph, an additional "grouping" column might be passed:

- "grouping" with the same value at each row with an exposure and exposure to include in a grouping. See details and examples below.

`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.

`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,

`returnSQL` Set to True to also return the rendered sql-query which creates the four central temptables, useful for debugging purposes.

Details

When mapping vocabularies e.g. from medDRA to SNOMED, it is possible that the mapping produces multiple terms. To create a chronograph for the joint SNOMED-terms, data collected from server for single exposures and outcomes should not be used for multiple exposures or multiple outcomes, as the same person might be counted multiple times. You should instead specify groupings in a column named "grouping" in the exposureOutcomePairs-argument table. See the examples.

Value

If returnSQL is False: A dataframe named results with observed and expected outcome counts in periods relative to the exposure initiation date, for each outcome and exposure. If returnSQL is True: A list containing 1) data frame results 2) The sql-query that creates the four central temptables, for debugging purposes.

Examples

```
# Collect data with a specific "background" (we don't use the default - i.e. the entire database as a baseline)

exposureOutcomePairs <- cbind.data.frame("exposureId"=c(40170549), "outcomeId"=c(321389))
exposure_background <- c(1321636, 40167333, 711452, 40173533, 981774, 19030860, 19037401, 925102, 1337068)
outcome_background <- c(73609, 134401,  74725, 375251, 199063, 381580, 4046804, 442094, 436883)

 output <- getChronographData(connectionDetails = connectionDetails,
                              oracleTempSchema = NULL,
                              cdmDatabaseSchema = "YOUR_DATABASE_SCHEMA",
                              cdmVersion = "5",
                              exposureIds = exposure_background,
                              outcomeIds = outcome_background,
                              exposureOutcomePairs = exposureOutcomePairs,
                              exposureDatabaseSchema = cdmDatabaseSchema,
                              exposureTable = "drug_era",
                              outcomeDatabaseSchema = cdmDatabaseSchema,
                              outcomeTable = "condition_era",
                              shrinkage = 0.5,
                              icPercentile = 0.025)

 plot(IcTemporalPatternDiscovery::plotChronograph(data=output, exposureId = 40170549, outcomeId = 321389, tit


# Collect data for multiple exposures and multiple outcomes (grouping 1), as well as for single rows (grouping 2

grouping_df <- cbind.data.frame("grouping"=1, setNames(expand.grid(c(40170549, 1321636), c(321389,  73609,  1
single_combinations_df <- cbind.data.frame("grouping"=2:3, "exposureId"=c(40170549, 1321636), "outcomeId"=c(
exposureOutcomePairs <- rbind.data.frame(grouping_df, single_combinations_df)

 output <- getChronographData(connectionDetails = connectionDetails,
                              oracleTempSchema = NULL,
                              cdmDatabaseSchema = "YOUR_DATABASE_SCHEMA",
                              cdmVersion = "5",
                              exposureOutcomePairs = exposureOutcomePairs,
                              exposureDatabaseSchema = cdmDatabaseSchema,
                              exposureTable = "drug_era",
                              outcomeDatabaseSchema = cdmDatabaseSchema,
                              outcomeTable = "condition_era",
                              shrinkage = 0.5,
                              icPercentile = 0.025)

 plot(IcTemporalPatternDiscovery::plotChronograph(data=output, exposureGrouping = 1, outcomeGrouping = 1, tit
```

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 = "5",
  exposureOutcomePairs,
  exposureDatabaseSchema = cdmDatabaseSchema,
  exposureTable = "drug_era",
  outcomeDatabaseSchema = cdmDatabaseSchema,
  outcomeTable = "condition_era",
  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 supports "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.

`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`.

`plotChronograph` *Plot a chronograph*

Description

Creates a plot showing the observed and expected nunber of outcomes for each month in the 3 years before and after initiation of the exposure, as well as the IC. The full process is described in Noren et al.

Usage

```
plotChronograph(
  data,
  exposureId,
  outcomeId,
  grouping = NULL,
  title = NULL,
  fileName = NULL
)
```

Arguments

`data`	Data as generated using the `getChronographData` function.
`exposureId`	The unique ID identifying the exposure to plot.
`outcomeId`	The unique ID identifying the outcome to plot.
`grouping`	The unique grouping ID, in case `getChronographData` was called with groupings.
`title`	The title to show above the plot.
`fileName`	Name of the file where the plot should be saved, for example 'plot.png'. See the function `ggsave` in the ggplot2 package for supported file formats.

References

Noren GN, Hopstadius J, Bate A, Star K, Edwards R, Temporal pattern discovery in longitudinal electronic patient records, Data Mining and Knowledge Discovery, May 2010, Volume 20, Issue 3, pp 361-387.

`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