Package 'MethodEvaluation'

April 13, 2023

Type Package

Title Package for Evaluation of Estimation Methods

Version 2.3.0

Date 2023-04-13

Maintainer Martijn Schuemie <schuemie@ohdsi.org>

Description Resources for the evaluation of the performance of methods that aim to estimate the magnitude (relative risk) of the effect of a drug on an outcome. These resources include reference sets for evaluating methods on real data, as well as functions for inserting simulated effects in real data based on negative control drug-outcome pairs. Further included are functions for the computation of the minimum detectable relative risks and functions for computing performance statistics such as predictive accuracy, error and bias.

License Apache License 2.0

VignetteBuilder knitr

URL https://github.com/OHDSI/MethodEvaluation

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

Depends R (>= 3.5.0),
DatabaseConnector (>= 6.0.0),
FeatureExtraction (>= 3.0.0),
Cyclops (>= 3.0.0)

Imports dplyr,
bit64,
readr,
purrr,
rlang,
SqlRender (>= 1.7.0),
pROC,
ggplot2,
ParallelLogger (>= 2.0.0),
methods,
EmpiricalCalibration,
checkmate

Suggests Eunomia,
testthat,
DT,

shiny,
knitr,
rmarkdown

Remotes ohdsi/FeatureExtraction,
ohdsi/Eunomia

RoxygenNote 7.2.3

Encoding UTF-8

R topics documented:

computeMdrr 2
computeMetrics 4
computeOhdsiBenchmarkMetrics 5
createReferenceSetCohorts 6
euadrReferenceSet 7
injectSignals 8
launchMethodEvaluationApp 11
ohdsiDevelopmentNegativeControls 12
ohdsiNegativeControls 13
omopReferenceSet 14
packageCustomBenchmarkResults 15
packageOhdsiBenchmarkResults 16
plotControls 17
plotCoverageInjectedSignals 18
plotRocsInjectedSignals 18
synthesizePositiveControls 19
synthesizeReferenceSetPositiveControls 22

Index **25**

`computeMdrr` *Compute minimal detectable relative risk (MDRR)*

Description

`computeMdrr` computes the minimal detectable relative risk (MDRR) for drug-outcome pairs using a standard approach that stratifies by age and gender (Armstrong 1987).

Usage

```
computeMdrr(
  connectionDetails,
  oracleTempSchema = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  cdmDatabaseSchema,
  exposureOutcomePairs,
  exposureDatabaseSchema = cdmDatabaseSchema,
  exposureTable = "drug_era",
  outcomeDatabaseSchema = cdmDatabaseSchema,
  outcomeTable = "condition_era",
```

```
  cdmVersion = "5"
)
```

Arguments

connectionDetails	An R object of type ConnectionDetails created using the function createConnectionDetails in the DatabaseConnector package.
oracleTempSchema	DEPRECATED: use 'tempEmulationSchema' instead.
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.
cdmDatabaseSchema	Name of database schema that contains OMOP CDM and vocabulary.
exposureOutcomePairs	A data frame with at least two columns: • "exposureId" or "targetId" containing the drug_concept_ID or cohort_definition_id of the exposure variable • "outcomeId" containing the condition_concept_ID or cohort_definition_id of the outcome variable
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".

Value

A data frame containing the MDRRs for the given exposure-outcome pairs.

References

Armstrong B. A simple estimator of minimum detectable relative risk, sample size, or power in cohort studies. American journal of epidemiology. 1987; 126: 356-8.

Examples

```
## Not run:
connectionDetails <- createConnectionDetails(
  dbms = "sql server",
  server = "RNDUSRDHIT07.jnj.com"
)
exposureOutcomePairs <- data.frame(
  exposureId = c(767410, 1314924, 907879),
  outcomeId = c(444382, 79106, 138825)
)
mdrrs <- computeMdrr(connectionDetails,
  "cdm_truven_mdcr",
  exposureOutcomePairs,
  outcomeTable = "condition_era"
)

## End(Not run)
```

`computeMetrics` *Compute method performance metrics*

Description

Compute method performance metrics

Usage

```
computeMetrics(
  logRr,
  seLogRr = NULL,
  ci95Lb = NULL,
  ci95Ub = NULL,
  p = NULL,
  trueLogRr
)
```

Arguments

`logRr`	A numeric vector of effect estimates on the log scale.
`seLogRr`	The standard error of the log of the effect estimates. Hint: often the standard error = (log(<lower bound 95 percent confidence interval>) - log(<effect estimate>))/qnorm(0.025). If not provided the standard error will be inferred from the 95 percent confidence interval.
`ci95Lb`	The lower bound of the 95 percent confidence interval. IF not provided it will be inferred from the standard error.
`ci95Ub`	The upper bound of the 95 percent confidence interval. IF not provided it will be inferred from the standard error.
`p`	The two-sided p-value corresponding to the null hypothesis of no effect. IF not provided it will be inferred from the standard error.
`trueLogRr`	A vector of the true effect sizes

Details

Compute the AUC, coverage, mean precision, MSE, type 1 error, type 2 error, and the fraction non-estimable.

Examples

```
library(EmpiricalCalibration)
data <- simulateControls(n = 50 * 3, trueLogRr = log(c(1, 2, 4)))
computeMetrics(logRr = data$logRr, seLogRr = data$seLogRr, trueLogRr = data$trueLogRr)
```

---

computeOhdsiBenchmarkMetrics

Generate perfomance metrics for the OHDSI Methods Benchmark

Description

Generate perfomance metrics for the OHDSI Methods Benchmark

Usage

```
computeOhdsiBenchmarkMetrics(
  exportFolder,
  mdrr = 1.25,
  stratum = "All",
  trueEffectSize = "Overall",
  calibrated = FALSE,
  comparative = FALSE
)
```

Arguments

exportFolder	The folder containing the CSV files created using the packageOhdsiBenchmarkResults function. This folder can contain results from various methods, analyses, and databases.
mdrr	The minimum detectable relative risk (MDRR). Only controls with this MDRR will be used to compute the performance metrics. Set to "All" to include all controls.
stratum	The stratum for which to compute the metrics, e.g. 'Acute Pancreatitis'. Set to 'All' to use all controls.
trueEffectSize	Should the analysis be limited to a specific true effect size? Set to "Overall" to include all.
calibrated	Should confidence intervals and p-values be empirically calibrated before computing the metrics?
comparative	Should the methods be evaluated on the task of comprative effect estimation? If FALSE, they will be evaluated on the task of effect estimation.

Value

A data frame with the various metrics per method - analysisId - database combination.

createReferenceSetCohorts *Create cohorts used in a reference set.*

Description

Create cohorts used in a reference set.

Usage

```
createReferenceSetCohorts(
  connectionDetails,
  oracleTempSchema = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  cdmDatabaseSchema,
  exposureDatabaseSchema = cdmDatabaseSchema,
  exposureTable = "exposures",
  outcomeDatabaseSchema = cdmDatabaseSchema,
  outcomeTable = "outcomes",
  nestingDatabaseSchema = cdmDatabaseSchema,
  nestingTable = "nesting",
  referenceSet = "ohdsiMethodsBenchmark",
  workFolder
)
```

Arguments

connectionDetails	An R object of type `ConnectionDetails` created using the function `createConnectionDetails` in the `DatabaseConnector` package.
oracleTempSchema	DEPRECATED: use ‘tempEmulationSchema‘ instead.
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.
cdmDatabaseSchema	A database schema containing health care data in the OMOP Commond Data Model. Note that for SQL Server, botth the database and schema should be specified, e.g. ’cdm_schema.dbo’.
exposureDatabaseSchema	The name of the database schema where the exposure cohorts will be created. Only needed if `referenceSet = 'ohdsiDevelopment'`. Note that for SQL Server, both the database and schema should be specified, e.g. ’cdm_schema.dbo’.
exposureTable	The name of the table that will be created to store the exposure cohorts. Only needed if `referenceSet = 'ohdsiDevelopment'`.
outcomeDatabaseSchema	The database schema where the target outcome table is located. Note that for SQL Server, both the database and schema should be specified, e.g. ’cdm_schema.dbo’

`outcomeTable` The name of the table where the outcomes will be stored.

`nestingDatabaseSchema`
(For the OHDSI Methods Benchmark and OHDSI Development Set only) The database schema where the nesting outcome table is located. Note that for SQL Server, both the database and schema should be specified, e.g. 'cdm_schema.dbo'.

`nestingTable` (For the OHDSI Methods Benchmark and OHDSI Development Set only) The name of the table where the nesting cohorts will be stored.

`referenceSet` The name of the reference set for which outcomes need to be created. Currently supported are "omopReferenceSet", "euadrReferenceSet", "ohdsiMethodsBenchmark", and "ohdsiDevelopment".

`workFolder` Name of local folder to place intermediary results; make sure to use forward slashes (/). Do not use a folder on a network drive since this greatly impacts performance.

Details

This function will create the outcomes of interest and nesting cohorts referenced in the various reference sets. The outcomes of interest are derives using information like diagnoses, procedures, and drug prescriptions. The outcomes are stored in a table on the database server.

For the 'ohdsiMethodsBenchmark' reference set, the exposures are taken from the drug_era table, and are therefore not generated as separate cohorts, and an exposure cohort table therefore needn't be supplied. For the 'ohdsiDevelopment' reference set, exposure cohorts will be generated.

`euadrReferenceSet` *The EU-ADR reference set A reference set of 43 drug-outcome pairs where we believe the drug causes the outcome (positive controls) and 50 drug-outcome pairs where we believe the drug does not cause the outcome (negative controls). The controls involve 10 health outcomes of interest. Note that originally, there was an additional positive control (Nimesulide and acute liver injury), but Nimesulide is not in RxNorm, and is not available in many countries.*

Description

The EU-ADR reference set A reference set of 43 drug-outcome pairs where we believe the drug causes the outcome (positive controls) and 50 drug-outcome pairs where we believe the drug does not cause the outcome (negative controls). The controls involve 10 health outcomes of interest. Note that originally, there was an additional positive control (Nimesulide and acute liver injury), but Nimesulide is not in RxNorm, and is not available in many countries.

Usage

```
data(euadrReferenceSet)
```

Format

A data frame with 399 rows and 10 variables:

exposureId Concept ID identifying the exposure

exposureName Name of the exposure

outcomeId Concept ID identifying the outcome

outcomeName Name of the outcome

groundTruth 0 = negative control, 1 = positive control

indicationId Concept Id identifying the (primary) indication of the drug. To be used when one wants to nest the analysis within the indication

indicationName Name of the indication

comparatorId Concept ID identifying a comparator drug that can be used as a counterfactual

comparatorName Name of the comparator drug

comparatorType How the comparator was selected

References

Coloma PM, Avillach P, Salvo F, Schuemie MJ, Ferrajolo C, Pariente A, Fourrier-Reglat A, Molokhia M, Patadia V, van der Lei J, Sturkenboom M, Trifiro G. A reference standard for evaluation of methods for drug safety signal detection using electronic healthcare record databases. Drug Safety 36(1):13-23, 2013

`injectSignals` *Inject signals in database*

Description

Inject signals in database

Usage

```
injectSignals(
  connectionDetails,
  cdmDatabaseSchema,
  oracleTempSchema = cdmDatabaseSchema,
  exposureDatabaseSchema = cdmDatabaseSchema,
  exposureTable = "drug_era",
  outcomeDatabaseSchema = cdmDatabaseSchema,
  outcomeTable = "cohort",
  outputDatabaseSchema = outcomeDatabaseSchema,
  outputTable = outcomeTable,
  createOutputTable = FALSE,
  exposureOutcomePairs,
  modelType = "poisson",
  minOutcomeCountForModel = 100,
  minOutcomeCountForInjection = 25,
 covariateSettings = FeatureExtraction::createCovariateSettings(useDemographicsAgeGroup
    = TRUE, useDemographicsGender = TRUE, useDemographicsIndexYear = TRUE,
    useDemographicsIndexMonth = TRUE, useConditionGroupEraLongTerm = TRUE,
    useDrugGroupEraLongTerm = TRUE, useProcedureOccurrenceLongTerm = TRUE,
  useMeasurementLongTerm = TRUE, useObservationLongTerm = TRUE, useCharlsonIndex =
  TRUE, useDcsi = TRUE, useChads2Vasc = TRUE, longTermStartDays = 365, endDays = 0),
 prior = Cyclops::createPrior("laplace", exclude = 0, useCrossValidation = TRUE),
 control = Cyclops::createControl(cvType = "auto", startingVariance = 0.1, noiseLevel =
```

```
    "quiet", threads = 10),
  firstExposureOnly = FALSE,
  washoutPeriod = 183,
  riskWindowStart = 0,
  riskWindowEnd = 0,
  addExposureDaysToEnd = TRUE,
  addIntentToTreat = FALSE,
  firstOutcomeOnly = FALSE,
  removePeopleWithPriorOutcomes = FALSE,
  maxSubjectsForModel = 1e+05,
  effectSizes = c(1, 1.25, 1.5, 2, 4),
  precision = 0.01,
  outputIdOffset = 1000,
  workFolder = "./SignalInjectionTemp",
  cdmVersion = "5",
  modelThreads = 1,
  generationThreads = 1
)
```

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.

`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 table name 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.

`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 table name that contains the outcome cohorts. When the table name is not CONDITION_ERA This table is expected to have the same format as the COHORT table: SUBJECT_ID, COHORT_START_DATE, COHORT_END_DATE, COHORT_CONCEPT_ID (CDM v4) or COHORT_DEFINITION_ID (CDM v5 and higher).

`outputDatabaseSchema`
: The name of the database schema that is the location of the tables containing the new outcomesRequires write permissions to this database.

`outputTable`
: The name of the table names that will contain the generated outcome cohorts.

createOutputTable
: Should the output table be created prior to inserting the outcomes? If TRUE and the tables already exists, it will first be deleted. If FALSE, the table is assumed to exist and the outcomes will be inserted. Any existing outcomes with the same IDs will first be deleted.

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

modelType
: Can be either "poisson" or "survival"

minOutcomeCountForModel
: Minimum number of outcome events required to build a model.

minOutcomeCountForInjection
: Minimum number of outcome events required to inject a signal.

covariateSettings
: An object of type `covariateSettings` as created using the `createCovariateSettings` function in the `FeatureExtraction` package.

prior
: The prior used to fit the outcome 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.

firstExposureOnly
: Should signals be injected only for the first exposure? (ie. assuming an acute effect)

washoutPeriod
: Number of days at the start of observation for which no signals will be injected, but will be used to determine whether exposure or outcome is the first one, and for extracting covariates to build the outcome model.

riskWindowStart
: The start of the risk window relative to the start of the exposure (in days). When 0, risk is assumed to start on the first day of exposure.

riskWindowEnd
: The end of the risk window relative to the start of the exposure. Note that typically the length of exposure is added to this number (when the `addExposureDaysToEnd` parameter is set to TRUE).

addExposureDaysToEnd
: Should length of exposure be added to the risk window?

addIntentToTreat
: If true, the signal will not only be injected in the primary time at risk, but also after the time at risk (up until the obseration period end). In both time periods, the target effect size will be enforced. This allows the same positive control synthesis to be used in both on treatment and intent-to-treat analysis variants. However, this will preclude the controls to be used in self-controlled designs that consider the time after exposure. Requires `firstExposureOnly = TRUE`.

firstOutcomeOnly
: Should only the first outcome per person be considered when modeling the outcome?

removePeopleWithPriorOutcomes
: Remove people with prior outcomes?

`maxSubjectsForModel`
: Maximum number of people used to fit an outcome model.

`effectSizes`
: A numeric vector of effect sizes that should be inserted.

`precision`
: The allowed ratio between target and injected signal size.

`outputIdOffset`
: What should be the first new outcome ID that is to be created?

`workFolder`
: Path to a folder where intermediate data will be stored.

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

`modelThreads`
: Number of parallel threads to use when fitting outcome models.

`generationThreads`
: Number of parallel threads to use when generating outcomes.

A data.frame listing all the drug-pairs in combination with requested effect sizes and the real inserted effect size (might be different from the requested effect size because of sampling error).

Details

DEPRECATED. Use `synthesizePositiveControls` instead.

---

launchMethodEvaluationApp

Launch the Method Evaluation Shiny app

Description

Launch the Method Evaluation Shiny app

Usage

```
launchMethodEvaluationApp(exportFolder, launch.browser = TRUE)
```

Arguments

`exportFolder`
: A folder where the data files for the Method Evaluation app are stored. Use the `packageOhdsiBenchmarkResults` function to populate this folder.

`launch.browser`
: Should the app be launched in your default browser, or in a Shiny window. Note: copying to clipboard will not work in a Shiny window.

Details

Launches a Shiny app that allows the user to explore the results.

ohdsiDevelopmentNegativeControls

The OHDSI Development Set - Negative Controls A set of 76 negative control outcomes, all for the exposures of ACE inhibitors (compared to thiazides and thiazide-like diuretics). This set is a much small set than the he OHDSI Method Evaluation Benchmark, but follows the same principles. It is intended to be used when developing methods, leaving the Methods Benchark untouched until a final evaluation of the method, thus preventing 'training' on the evaluation set. The negative controls are borrowed from the LEGEND Hypertension study. The exposure, outcome, and nesting cohorts can be created using the `createReferenceSetCohorts` *function. These negative controls can form the basis to generate positive controls using the* `injectSignals` *function.*

Description

The OHDSI Development Set - Negative Controls A set of 76 negative control outcomes, all for the exposures of ACE inhibitors (compared to thiazides and thiazide-like diuretics). This set is a much small set than the he OHDSI Method Evaluation Benchmark, but follows the same principles. It is intended to be used when developing methods, leaving the Methods Benchark untouched until a final evaluation of the method, thus preventing 'training' on the evaluation set. The negative controls are borrowed from the LEGEND Hypertension study. The exposure, outcome, and nesting cohorts can be created using the `createReferenceSetCohorts` function. These negative controls can form the basis to generate positive controls using the `injectSignals` function.

Usage

```
data(ohdsiDevelopmentNegativeControls)
```

Format

A data frame with 76 rows and 11 variables:

targetId Cohort ID identifying the target exposure

targetName Name of the target cohort

comparatorId Cohort ID identifying the comparator exposure

comparatorName Name of the comparator cohort

nestingId Cohort ID identifying the nesting cohort

nestingName Name of the nesting cohort

outcomeId Cohort ID identifying the outcome

outcomeName Name of the outcome

type THe type of control: exposure or outcome

targetConceptIds A semi-colon separated list of concept IDs that together form the target exposure definitions.

comparatorConceptIds A semi-colon separated list of concept IDs that together form the comparator exposure definitions.

`ohdsiNegativeControls` *The OHDSI Method Evaluation Benchmark - Negative Controls A set of 200 negative controls, centered around four outcomes of interest (acute pancreatitis, GI bleeding, Stroke, and IBD), and 4 exposures of interest (diclofenac, ciprofloxacin, metformin, and sertraline), which 25 negative controls each. Each drug-outcome pair also includes a comparator drug (where the comparator is also a negative control), allowing for evaluation of comparative effect estimation, and a nesting cohort for evaluating methods such as the nested case-control design. The exposure, outcome, and nesting cohorts can be created using the* `createReferenceSetCohorts` *function. These negative controls can form the basis to generate positive controls using the* `injectSignals` *function.*

Description

The OHDSI Method Evaluation Benchmark - Negative Controls A set of 200 negative controls, centered around four outcomes of interest (acute pancreatitis, GI bleeding, Stroke, and IBD), and 4 exposures of interest (diclofenac, ciprofloxacin, metformin, and sertraline), which 25 negative controls each. Each drug-outcome pair also includes a comparator drug (where the comparator is also a negative control), allowing for evaluation of comparative effect estimation, and a nesting cohort for evaluating methods such as the nested case-control design. The exposure, outcome, and nesting cohorts can be created using the `createReferenceSetCohorts` function. These negative controls can form the basis to generate positive controls using the `injectSignals` function.

Usage

```
data(ohdsiNegativeControls)
```

Format

A data frame with 200 rows and 9 variables:

targetId Cohort ID identifying the target exposure

targetName Name of the target cohort

comparatorId Cohort ID identifying the comparator exposure

comparatorName Name of the comparator cohort

nestingId Cohort ID identifying the nesting cohort

nestingName Name of the nesting cohort

outcomeId Cohort ID identifying the outcome

outcomeName Name of the outcome

type THe type of control: exposure or outcome

targetConceptIds A semi-colon separated list of concept IDs that together form the target exposure definitions.

comparatorConceptIds A semi-colon separated list of concept IDs that together form the comparator exposure definitions.

`omopReferenceSet` *The OMOP reference set A reference set of 165 drug-outcome pairs where we believe the drug causes the outcome (positive controls) and 234 drug-outcome pairs where we believe the drug does not cause the outcome (negative controls). The controls involve 4 health outcomes of interest: acute liver injury, acute kidney injury, acute myocardial infarction, and GI bleeding.*

Description

The OMOP reference set A reference set of 165 drug-outcome pairs where we believe the drug causes the outcome (positive controls) and 234 drug-outcome pairs where we believe the drug does not cause the outcome (negative controls). The controls involve 4 health outcomes of interest: acute liver injury, acute kidney injury, acute myocardial infarction, and GI bleeding.

Usage

```
data(omopReferenceSet)
```

Format

A data frame with 399 rows and 10 variables:

exposureId Concept ID identifying the exposure

exposureName Name of the exposure

outcomeId Concept ID identifying the outcome

outcomeName Name of the outcome

groundTruth 0 = negative control, 1 = positive control

indicationId Concept Id identifying the (primary) indication of the drug. To be used when one wants to nest the analysis within the indication

indicationName Name of the indication

comparatorId Concept ID identifying a comparator drug that can be used as a counterfactual

comparatorName Name of the comparator drug

comparatorType How the comparator was selected

References

Ryan PB, Schuemie MJ, Welebob E, Duke J, Valentine S, Hartzema AG. Defining a reference set to support methodological research in drug safety. Drug Safety 36 Suppl 1:S33-47, 2013

`packageCustomBenchmarkResults` *Package results of a method on the OHDSI Methods Benchmark*

Description

Stores the results of a method on the OHDSI Methods Benchmark in a standardized format, for example for use in the Method Evaluation Shiny app.

Usage

```
packageCustomBenchmarkResults(
  estimates,
  negativeControls,
  synthesisSummary,
  analysisRef,
  databaseName,
  exportFolder
)
```

Arguments

`estimates`	A data frame containing the estimates. See details for required columns.
`negativeControls`	A data frame containing the negative controls. See details for required columns.
`synthesisSummary`	A data frame with the summary of the positive control synthesis as generated by the `synthesizePositiveControls` function.
`analysisRef`	A file describing the various analyses that were performed. See details for required columns.
`databaseName`	A character string to identify the database the method was executed on.
`exportFolder`	The folder where the output CSV files will written.

Details

The `estimates` argument should have the following columns: "targetId", "outcomeId", "analysisId", "logRr", "seLogRr", "ci95Lb", "ci95Ub". The `negativeControls` argument should have the following columns: "targetId", "outcomeId", "type". The `analysisRef` argument should have the following columns: "analysisId", "method", "comparative", "nesting", "firstExposureOnly" The `targetId` and `outcomeId` fields identify the specific control, and should correspond to those in the negativeControls and synthesisSummary objects. The `type` fields should be either "Outcome control" or "Exposure control". The `analysisId` field is an integer that identifies a specific variant of the method. For example, if the method is 'CohortMethod', analysisId = 1 could identify a set of settings using propensity score matching, and analysisId = 2 could identify a set of settings using stratification. `logRr`, `seLogRr`, `ci95Lb`, and `ci95Ub` correspond to the log of the effect estimate (e.g. the hazard ratio), the standard error, and the upper and lower bound of the effect size estimate, as produced by the method. `method` is a character string identifyign the method (e.g. "CohortMethod"). `comparative` is a boolean indicating whether the analysis can also be considerd to perform comparative effect estimation (comparing the target to the comparator). `nesting` is a

boolean indicating whether the analysis is nested in the nesting cohorts identified in the gold standard. `firstExposureOnly` is a boolean indicating whether only the first exposure was used in the analysis.

`packageOhdsiBenchmarkResults` *Package results of a method on the OHDSI Methods Benchmark*

Description

Stores the results of a method on the OHDSI Methods Benchmark in a standardized format, for example for use in the Method Evaluation Shiny app.

Usage

```
packageOhdsiBenchmarkResults(
  estimates,
  controlSummary,
  analysisRef,
  databaseName,
  exportFolder,
  referenceSet = "ohdsiMethodsBenchmark"
)
```

Arguments

`estimates`	A data frame containing the estimates. See details for required columns.
`controlSummary`	A data frame with the summary of the controls as generated by the `synthesizeReferenceSetPositi` function.
`analysisRef`	A file describing the various analyses that were performed. See details for required columns.
`databaseName`	A character string to identify the database the method was executed on.
`exportFolder`	The folder where the output CSV files will written.
`referenceSet`	The name of the reference set for which to package the results. Currently supported are "ohdsiMethodsBenchmark" and "ohdsiDevelopment".

Details

The `estimates` argument should have the following columns: "targetId", "outcomeId", "analysisId", "logRr", "seLogRr", "ci95Lb", "ci95Ub". The `analysisRef` argument should have the following columns: "analysisId", "method", "comparative", "nesting", "firstExposureOnly" The `targetId` and `outcomeId` fields identify the specific control, and should correspond to those in the controlSummary object. The `analysisId` field is an integer that identifies a specific variant of the method. For example, if the method is 'CohortMethod', analysisId = 1 could identify a set of settings using propensity score matching, and analysisId = 2 could identify a set of settings using stratification. `logRr`, `seLogRr`, `ci95Lb`, and `ci95Ub` correspond to the log of the effect estimate (e.g. the hazard ratio), the standard error, and the upper and lower bound of the effect size estimate, as produced by the method. `method` is a character string identifyign the method (e.g. "CohortMethod"). `comparative` is a boolean indicating whether the analysis can also be considerd to perform comparative effect estimation (comparing the target to the comparator). `nesting` is a

boolean indicating whether the analysis is nested in the nesting cohorts identified in the gold standard. `firstExposureOnly` is a boolean indicating whether only the first exposure was used in the analysis.

`plotControls` *Plot negative and positive control estimates.*

Description

Plot negative and positive control estimates.

Usage

```
plotControls(
  logRr,
  seLogRr = NULL,
  ci95Lb = NULL,
  ci95Ub = NULL,
  trueLogRr,
  estimateType = "relative risk",
  fileName = NULL,
  title = NULL
)
```

Arguments

`logRr`	A numeric vector of effect estimates on the log scale.
`seLogRr`	The standard error of the log of the effect estimates. Hint: often the standard error = (log(<lower bound 95 percent confidence interval>) - log(<effect estimate>))/qnorm(0.025). If not provided the standard error will be inferred from the 95 percent confidence interval.
`ci95Lb`	The lower bound of the 95 percent confidence interval. IF not provided it will be inferred from the standard error.
`ci95Ub`	The upper bound of the 95 percent confidence interval. IF not provided it will be inferred from the standard error.
`trueLogRr`	A vector of the true effect sizes
`estimateType`	A character string to denote the effect size estimate type. Used for the x-axis and the true effect size labels.
`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.
`title`	An optional title to display above the plot.

Value

A Ggplot object. Use the `ggsave` function to save to file.

plotCoverageInjectedSignals *Plot the coverage*

Description

Plot the coverage

Usage

```
plotCoverageInjectedSignals(
  logRr,
  seLogRr,
  trueLogRr,
  region = 0.95,
  fileName = NULL
)
```

Arguments

logRr	A numeric vector of effect estimates on the log scale
seLogRr	The standard error of the log of the effect estimates. Hint: often the standard error = (log(<lower bound 95 percent confidence interval>) - log(<effect estimate>))/qnorm(0.025)
trueLogRr	A vector of the true effect sizes
region	Size of the confidence interval. Default is .95 (95 percent).
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.

Details

Plot the fractions of estimates where the true effect size is below, above or within the confidence interval, for one or more true effect sizes.

plotRocsInjectedSignals *Plot the ROC curves for various injected signal sizes*

Description

Plot the ROC curves for various injected signal sizes

Usage

```
plotRocsInjectedSignals(logRr, trueLogRr, showAucs, fileName = NULL)
```

Arguments

logRr	A vector containing the log of the relative risk as estimated by a method.
trueLogRr	A vector containing the injected log(relative risk) for each estimate.
showAucs	Should the AUCs be shown in 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.

Value

A Ggplot object. Use the ggsave function to save to file.

---

synthesizePositiveControls
Synthesize positive controls

Description

Synthesize positive controls

Usage

```
synthesizePositiveControls(
  connectionDetails,
  cdmDatabaseSchema,
  oracleTempSchema = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  exposureDatabaseSchema = cdmDatabaseSchema,
  exposureTable = "drug_era",
  outcomeDatabaseSchema = cdmDatabaseSchema,
  outcomeTable = "cohort",
  outputDatabaseSchema = outcomeDatabaseSchema,
  outputTable = outcomeTable,
  createOutputTable = FALSE,
  exposureOutcomePairs,
  modelType = "poisson",
  minOutcomeCountForModel = 100,
  minOutcomeCountForInjection = 25,
  minModelCount = 5,
  covariateSettings = FeatureExtraction::createCovariateSettings(useDemographicsAgeGroup
    = TRUE, useDemographicsGender = TRUE, useDemographicsIndexYear = TRUE,
    useDemographicsIndexMonth = TRUE, useConditionGroupEraLongTerm = TRUE,
    useDrugGroupEraLongTerm = TRUE, useProcedureOccurrenceLongTerm = TRUE,
   useMeasurementLongTerm = TRUE, useObservationLongTerm = TRUE, useCharlsonIndex =
   TRUE, useDcsi = TRUE, useChads2Vasc = TRUE, longTermStartDays = 365, endDays = 0),
  prior = Cyclops::createPrior("laplace", exclude = 0, useCrossValidation = TRUE),
  control = Cyclops::createControl(cvType = "auto", startingVariance = 0.1, seed = 1,
    resetCoefficients = TRUE, noiseLevel = "quiet", threads = 10),
  firstExposureOnly = FALSE,
  washoutPeriod = 183,
  riskWindowStart = 0,
```

```
  riskWindowEnd = 0,
  endAnchor = "cohort end",
  addIntentToTreat = FALSE,
  firstOutcomeOnly = FALSE,
  removePeopleWithPriorOutcomes = FALSE,
  maxSubjectsForModel = 1e+05,
  effectSizes = c(1, 1.25, 1.5, 2, 4),
  precision = 0.01,
  outputIdOffset = 1000,
  workFolder = "./SignalInjectionTemp",
  cdmVersion = "5",
  modelThreads = 1,
  generationThreads = 1
)
```

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`
: DEPRECATED: use 'tempEmulationSchema' instead.

`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 by assumed to be cdmSchema. Requires read permissions to this database.

`exposureTable`
: The table name 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.

`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 table name that contains the outcome cohorts. When the table name is not CONDITION_ERA This table is expected to have the same format as the COHORT table: SUBJECT_ID, COHORT_START_DATE, COHORT_END_DATE, COHORT_CONCEPT_ID (CDM v4) or COHORT_DEFINITION_ID (CDM v5 and higher).

`outputDatabaseSchema`
: The name of the database schema that is the location of the tables containing the new outcomesRequires write permissions to this database.

`outputTable`
: The name of the table names that will contain the generated outcome cohorts.

`createOutputTable`
: Should the output table be created prior to inserting the outcomes? If TRUE and the tables already exists, it will first be deleted. If FALSE, the table is assumed to exist and the outcomes will be inserted. Any existing outcomes with the same IDs will first be deleted.

`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

`modelType`
: Can be either "poisson" or "survival"

`minOutcomeCountForModel`
: Minimum number of outcome events required to build a model.

`minOutcomeCountForInjection`
: Minimum number of outcome events required to inject a signal.

`minModelCount`
: Minimum number of negative controls having enough outcomes to fit an outcome model.

`covariateSettings`
: An object of type `covariateSettings` as created using the `createCovariateSettings` function in the `FeatureExtraction` package.

`prior`
: The prior used to fit the outcome 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.

`firstExposureOnly`
: Should signals be injected only for the first exposure? (ie. assuming an acute effect)

`washoutPeriod`
: Number of days at the start of observation for which no signals will be injected, but will be used to determine whether exposure or outcome is the first one, and for extracting covariates to build the outcome model.

`riskWindowStart`
: The start of the risk window relative to the start of the exposure (in days). When 0, risk is assumed to start on the first day of exposure.

`riskWindowEnd`
: The end of the risk window (in days) relative to the endAnchor.

`endAnchor`
: The anchor point for the end of the risk window. Can be "cohort start" or "cohort end".

`addIntentToTreat`
: If true, the signal will not only be injected in the primary time at risk, but also after the time at risk (up until the obseration period end). In both time periods, the target effect size will be enforced. This allows the same positive control synthesis to be used in both on treatment and intent-to-treat analysis variants. However, this will preclude the controls to be used in self-controlled designs that consider the time after exposure. Requires `firstExposureOnly = TRUE`.

`firstOutcomeOnly`
: Should only the first outcome per person be considered when modeling the outcome?

`removePeopleWithPriorOutcomes`
: Remove people with prior outcomes?

`maxSubjectsForModel`
: Maximum number of people used to fit an outcome model.

`effectSizes`
: A numeric vector of effect sizes that should be inserted.

`precision`
: The allowed ratio between target and injected signal size.

`outputIdOffset`
: What should be the first new outcome ID that is to be created?

`workFolder`
: Path to a folder where intermediate data will be stored.

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

`modelThreads`
: Number of parallel threads to use when fitting outcome models.

`generationThreads`
: Number of parallel threads to use when generating outcomes.

Details

This function will insert additional outcomes for a given set of drug-outcome pairs. It is assumed that these drug-outcome pairs represent negative controls, so the true relative risk before inserting any outcomes should be 1. There are two models for inserting the outcomes during the specified risk window of the drug: a Poisson model assuming multiple outcomes could occurr during a single exposure, and a survival model considering only one outcome per exposure. It is possible to use bulk import to insert the generated outcomes in the database. This requires the environmental variable 'USE_MPP_BULK_LOAD' to be set to 'TRUE'. See `?DatabaseConnector::insertTable` for details on how to configure the bulk upload.

Value

A data.frame listing all the drug-pairs in combination with requested effect sizes and the real inserted effect size (might be different from the requested effect size because of sampling error).

References

Schuemie MJ, Hripcsak G, Ryan PB, Madigan D, Suchard MA. Empirical confidence interval calibration for population-level effect estimation studies in observational healthcare data. Proc Natl Acad Sci U S A. 2018 Mar 13;115(11):2571-2577.

`synthesizeReferenceSetPositiveControls` *Synthesize positive controls for reference set*

Description

Synthesize positive controls for reference set

Usage

```
synthesizeReferenceSetPositiveControls(
  connectionDetails,
  oracleTempSchema = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  cdmDatabaseSchema,
  exposureDatabaseSchema = cdmDatabaseSchema,
  exposureTable = "drug_era",
```

```
  outcomeDatabaseSchema = cdmDatabaseSchema,
  outcomeTable = "cohort",
  referenceSet = "ohdsiMethodsBenchmark",
  maxCores = 1,
  workFolder,
  summaryFileName = file.path(workFolder, "allControls.csv")
)
```

Arguments

connectionDetails	An R object of type ConnectionDetails created using the function createConnectionDetails in the DatabaseConnector package.
oracleTempSchema	DEPRECATED: use 'tempEmulationSchema' instead.
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.
cdmDatabaseSchema	A database schema containing health care data in the OMOP Commond Data Model. Note that for SQL Server, botth the database and schema should be specified, e.g. 'cdm_schema.dbo'
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 and assumed to be cdmDatabaseSchema. 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 database schema where the target outcome table is located. Note that for SQL Server, both the database and schema should be specified, e.g. 'cdm_schema.dbo'
outcomeTable	The name of the table where the outcomes will be stored.
referenceSet	The name of the reference set for which positive controls need to be synthesized. Currently supported are "ohdsiMethodsBenchmark" and "ohdsiDevelopment".
maxCores	How many parallel cores should be used? If more cores are made available this can speed up the analyses.
workFolder	Name of local folder to place intermediary results; make sure to use forward slashes (/). Do not use a folder on a network drive since this greatly impacts performance.
summaryFileName	The name of the CSV file where to store the summary of the final set of positive and negative controls.

Details

This function will synthesize positive controls for a given reference set based on the real negative controls. Data from the database will be used to fit outcome models for each negative control

outcome, and these models will be used to sample additional synthetic outcomes during exposure to increase the true hazard ratio. The positive control outcome cohorts will be stored in the same database table as the negative control outcome cohorts. A summary file will be created listing all positive and negative controls. This list should then be used as input for the method under evaluation.

Index

∗ **datasets**
euadrReferenceSet, 7
ohdsiDevelopmentNegativeControls, 12
ohdsiNegativeControls, 13
omopReferenceSet, 14

computeMdrr, 2
computeMetrics, 4
computeOhdsiBenchmarkMetrics, 5
createControl, *10*, *21*
createPrior, *10*, *21*
createReferenceSetCohorts, 6, *12*, *13*

euadrReferenceSet, 7

injectSignals, 8, *12*, *13*

launchMethodEvaluationApp, 11

ohdsiDevelopmentNegativeControls, 12
ohdsiNegativeControls, 13
omopReferenceSet, 14

packageCustomBenchmarkResults, 15
packageOhdsiBenchmarkResults, *5*, *11*, 16
plotControls, 17
plotCoverageInjectedSignals, 18
plotRocsInjectedSignals, 18

synthesizePositiveControls, *11*, *15*, 19
synthesizeReferenceSetPositiveControls, *16*, 22