Package 'PheValuator'

November 4, 2026

Type Package

Title Rapid Phenotype Evaluation

Version 2.2.19

Date 2026-10-7

Description Evaluates phenotype algorithms by developing a diagnostic prediction model using a set of noisy positives and noisy negatives. It can then apply the model to a large cohort to create a set of subjects each with a predicted probability for the health outcome of interest. This cohort can then be used to evaluate phenotype algorithms through the calculation of the necessary performance characteristics - sensitivity, specificity, and positive and negative predictive value. Based on work by Swerdel (2022) <doi:10.1016/j.jbi.2022.104177> and Swerdel (2019) <doi:10.1016/j.jbi.2019.103258>.

License Apache License 2.0

VignetteBuilder knitr

Depends R (>= 4.1.1)

Imports DatabaseConnector, PatientLevelPrediction (>= 6.0.4), FeatureExtraction (>= 2.1.5), SqlRender (>= 1.5.2), dplyr (>= 0.7.8), methods, ParallelLogger, tibble

Suggests knitr, rmarkdown, shiny, shinydashboard, DT, htmltools, qpdf, testthat, Eunomia, withr, CohortGenerator

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

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

Encoding UTF-8

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author William J. O'Brien [aut, cre], Joel Swerdel [aut], Martijn Schuemie [aut]

Maintainer William J. O'Brien <wobrien8@its.jnj.com>

Repository CRAN

Date/Publication 2026-11-04 22:10:08 UTC

Contents

computePerformanceMetricsFromCounts . . . 2
createCreateEvaluationCohortArgs . . . 3
createDefaultCovariateSettings . . . 6
createEvaluationCohort . . . 7
createPheValuatorAnalysis . . . 12
createTestPhenotypeAlgorithmArgs . . . 13
loadPheValuatorAnalysisList . . . 14
runPheValuatorAnalyses . . . 15
savePheValuatorAnalysisList . . . 17
summarizePheValuatorAnalyses . . . 18
testPhenotypeAlgorithm . . . 19

Index **22**

`computePerformanceMetricsFromCounts` *Compute performance metrics based on a 2-by-2 counts table*

Description

Compute performance metrics based on a 2-by-2 counts table

Usage

```
computePerformanceMetricsFromCounts(counts)
```

Arguments

`counts`	Counts as created by the `testPhenotypeAlgorithm` function.

Value

A tibble with the statistics metrics added to the counts table.

Examples

```
counts <- tibble::tibble(
  truePositives = 120,
  trueNegatives = 800,
  falsePositives = 30,
  falseNegatives = 50,
  cutPoint = "EV"
)
computePerformanceMetricsFromCounts(counts)
```

---

createCreateEvaluationCohortArgs *Create a parameter object for the function createEvaluationCohort*

Description

Create a parameter object for the function createEvaluationCohort

Usage

```
createCreateEvaluationCohortArgs(
  xSpecCohortId,
  daysFromxSpec = 0,
  xSensCohortId = prevalenceCohortId,
  prevalenceCohortId,
  caseCohortId = prevalenceCohortId,
  caseFirstOccurrenceOnly = TRUE,
  xSpecCohortSize = 5000,
 covariateSettings = createDefaultCovariateSettings(excludedCovariateConceptIds = c(),
    addDescendantsToExclude = TRUE),
  modelPopulationCohortId = 0,
  modelPopulationCohortIdStartDay = 0,
  modelPopulationCohortIdEndDay = 0,
  inclusionEvaluationCohortId = 0,
  inclusionEvaluationDaysFromStart = 0,
  inclusionEvaluationDaysFromEnd = 0,
  duringInclusionEvaluationOnly = FALSE,
  exclusionEvaluationCohortId = 0,
  exclusionEvaluationDaysFromStart = 0,
  exclusionEvaluationDaysFromEnd = 0,
  minimumOffsetFromStart = 0,
  minimumOffsetFromEnd = 30,
  modelBaseSampleSize = 25000,
  baseSampleSize = 2e+06,
  lowerAgeLimit = 0,
  upperAgeLimit = 120,
```

```
  visitLength = 0,
  visitType = c(9201, 9202, 9203, 581477, 262),
  gender = c(8507, 8532),
  race = 0,
  ethnicity = 0,
  startDate = "19001010",
  endDate = "21000101",
  falsePositiveNegativeSubjects = 10,
  modelId = "main",
  evaluationCohortId = "main",
  randomVisitTable = "",
  excludeModelFromEvaluation = FALSE,
  removeSubjectsWithFutureDates = TRUE,
  saveEvaluationCohortPlpData = FALSE
)
```

Arguments

`xSpecCohortId`	The number of the "extremely specific (xSpec)" cohortdefinition id in the cohort table (for noisy positives).
`daysFromxSpec`	Number of days from xSpec condition until analyzed visit
`xSensCohortId`	The number of the "extremely sensitive (xSens)" cohortdefinition id in the cohort table (for noisy negatives).
`prevalenceCohortId`	The number of the cohort definition id to determine the disease prevalence.
`caseCohortId`	The number of the cohort definition id to determine cases in the evaluation cohort
`caseFirstOccurrenceOnly`	Set to true if only the first occurrence per subject in the case cohort is to be used
`xSpecCohortSize`	Maximum xSpec sample size to use in the analysis
`covariateSettings`	A covariateSettings object as generated using createCovariateSettings().
`modelPopulationCohortId`	The number of the cohort to be used as a base population for the model. If set to 0, the entire database population will be used.
`modelPopulationCohortIdStartDay`	The number of days relative to the mainPopulationCohortId cohort start date to begin including visits.
`modelPopulationCohortIdEndDay`	The number of days relative to the mainPopulationCohortId cohort start date to end including visits.
`inclusionEvaluationCohortId`	The number of the cohort of the population to be used to designate which visits are are eligible to be in the evaluation cohort

`inclusionEvaluationDaysFromStart`
: The number of days from the cohort start date of the inclusionEvaluationCohortId to start eligible included visits

`inclusionEvaluationDaysFromEnd`
: The number of days from the cohort start date of the inclusionEvaluationCohortId to end eligible included visits

`duringInclusionEvaluationOnly`
: Only include visits that are within the cohort start and end dates

`exclusionEvaluationCohortId`
: The number of the cohort of the population to be used to designate which visits are NOT eligible to be in the evaluation cohort

`exclusionEvaluationDaysFromStart`
: The number of days from the cohort start date of the exclusionEvaluationCohortId to start ineligible included visits

`exclusionEvaluationDaysFromEnd`
: The number of days from the cohort start date of the exclusionEvaluationCohortId to end ineligible included visits

`minimumOffsetFromStart`
: Minimum number of days to offset for the analysis visit from the start of the observation period

`minimumOffsetFromEnd`
: Minimum number of days to offset for the analysis visit from the end of the observation period

`modelBaseSampleSize`
: The number of non-xSpec subjects to include in the model

`baseSampleSize` The maximum number of subjects in the evaluation cohort.

`lowerAgeLimit` The lower age for subjects in the model.

`upperAgeLimit` The upper age for subjects in the model.

`visitLength` The minimum length of index visit for acute outcomes.

`visitType` The concept_id for the visit type.

`gender` The gender(s) to be included.

`race` The race(s) to be included.

`ethnicity` The ethnicity(s) to be included.

`startDate` The starting date for including subjects in the model.

`endDate` The ending date for including subjects in the model.

`falsePositiveNegativeSubjects`
: Number of subjects to include for evaluating false positives and negatives

`modelId` A string used to generate the file names for this model.

`evaluationCohortId`
: A string used to generate the file names for this evaluation cohort.

`randomVisitTable`
: Table stored in work directory with pre-selected random visits in format of visit_occurrence table

`excludeModelFromEvaluation`
: Should subjects used in the model be excluded from the evaluation cohort?

`removeSubjectsWithFutureDates`
: For buggy data with data in the future: ignore subjects with dates in the future?

`saveEvaluationCohortPlpData`
: Should the large PLP file for the evaluation cohort be saved? To be used for debugging purposes.

Details

Create an object defining the parameter values.

Value

A list of class `"args"` containing all the provided parameters. This object is used as an argument to the `createEvaluationCohort` function.

Examples

```
evalCohortArgs <- createCreateEvaluationCohortArgs(
  xSpecCohortId = 1,
  prevalenceCohortId = 3
)
```

---

createDefaultCovariateSettings

Create the default covariate settings

Description

Create the default covariate settings for acute or chronic diseases

Usage

```
createDefaultCovariateSettings(
  excludedCovariateConceptIds = c(),
  includedCovariateIds = c(),
  includedCovariateConceptIds = c(),
  addDescendantsToExclude = FALSE,
  startDayWindow1 = 0,
  endDayWindow1 = 9999,
  startDayWindow2 = NULL,
  endDayWindow2 = NULL,
  startDayWindow3 = NULL,
  endDayWindow3 = NULL
)
```

Arguments

`excludedCovariateConceptIds`
: A list of conceptIds to exclude from featureExtraction. These should include all concept_ids that were used to define the xSpec model (default=NULL)

`includedCovariateIds`
: A list of covariate IDs that should be restricted to.

`includedCovariateConceptIds`
: A list of covariate concept IDs that should be restricted to.

`addDescendantsToExclude`
: Should descendants of excluded concepts also be excluded? (default=FALSE)

`startDayWindow1`
: The day to start time window 1 for feature extraction

`endDayWindow1`
: The day to end time window 1 for feature extraction

`startDayWindow2`
: The day to start time window 2 for feature extraction

`endDayWindow2`
: The day to end time window 2 for feature extraction

`startDayWindow3`
: The day to start time window 3 for feature extraction

`endDayWindow3`
: The day to end time window 3 for feature extraction #'

Details

Function to create the default covariate settings for acute or chronic diseases

Value

A list containing one or more covariate settings objects from the FeatureExtraction package. If only the first time window is specified, returns a list with one element. If additional time windows are specified, returns a list with multiple elements.

Examples

```
covSettings <- createDefaultCovariateSettings(
  excludedCovariateConceptIds = c(201826, 44786627),
  addDescendantsToExclude = TRUE
)
```

---

createEvaluationCohort

Create the evaluation cohort

Description

Create the evaluation cohort

Usage

```
createEvaluationCohort(
  connectionDetails,
  oracleTempSchema = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  phenotype,
  analysisName,
  runDateTime,
  databaseId,
  xSpecCohortId,
  daysFromxSpec = 0,
  xSensCohortId,
  prevalenceCohortId,
  caseCohortId,
  caseFirstOccurrenceOnly,
  xSpecCohortSize = 5000,
  cdmDatabaseSchema,
  cohortDatabaseSchema,
  cohortTable,
  workDatabaseSchema,
 covariateSettings = createDefaultCovariateSettings(excludedCovariateConceptIds = c(),
    addDescendantsToExclude = TRUE),
  modelPopulationCohortId = 0,
  modelPopulationCohortIdStartDay = 0,
  modelPopulationCohortIdEndDay = 0,
  inclusionEvaluationCohortId = 0,
  inclusionEvaluationDaysFromStart = 0,
  inclusionEvaluationDaysFromEnd = 0,
  duringInclusionEvaluationOnly = FALSE,
  exclusionEvaluationCohortId = 0,
  exclusionEvaluationDaysFromStart = 0,
  exclusionEvaluationDaysFromEnd = 0,
  priorModelToUse = NULL,
  minimumOffsetFromStart = 365,
  minimumOffsetFromEnd = 365,
  modelBaseSampleSize = 25000,
  baseSampleSize = 2e+06,
  lowerAgeLimit = 0,
  upperAgeLimit = 120,
  visitLength = 0,
  visitType = c(9201, 9202, 9203, 262, 581477),
  gender = c(8507, 8532),
  race = 0,
  ethnicity = 0,
  startDate = "19001010",
  endDate = "21000101",
  falsePositiveNegativeSubjects = 10,
  cdmVersion = "5",
```

```
  outFolder,
  exportFolder,
  modelId = "main",
  evaluationCohortId = "main",
  excludeModelFromEvaluation = FALSE,
  randomVisitTable = "",
  removeSubjectsWithFutureDates = TRUE,
  saveEvaluationCohortPlpData = FALSE
)
```

Arguments

connectionDetails	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.
phenotype	Name of the phenotype for analysis
analysisName	Name of the analysis
runDateTime	Starting date and time of the PheValuator run
databaseId	Name of the database in the analysis
xSpecCohortId	The number of the "extremely specific (xSpec)" cohort definition id in the cohort table (for noisy positives).
daysFromxSpec	Number of days allowed from xSpec condition until analyzed visit
xSensCohortId	The number of the "extremely sensitive (xSens)" cohort definition id in the cohort table (for noisy negatives).
prevalenceCohortId	The number of the cohort definition id to determine the disease prevalence.
caseCohortId	The number of the cohort definition id to determine cases in the evaluation cohort
caseFirstOccurrenceOnly	Set to true if only the first occurrence per subject in the case cohort is to be used
xSpecCohortSize	The recommended xSpec sample size to use in model (default = NULL)
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'.
cohortDatabaseSchema	The name of the database schema that is the location where the cohort data used to define the at risk cohort is available. Requires read permissions to this database.

`cohortTable` The tablename that contains the at risk cohort. The expectation is cohortTable has format of COHORT table: cohort_concept_id, SUBJECT_ID, COHORT_START_DATE, COHORT_END_DATE.

`workDatabaseSchema`
The name of the database schema that is the location where a table can be created and afterwards removed. Requires write permissions to this database.

`covariateSettings`
A covariateSettings object as generated using createCovariateSettings().

`modelPopulationCohortId`
The number of the cohort to be used as a base population for the model. If set to 0, the entire database population will be used.

`modelPopulationCohortIdStartDay`
The number of days relative to the mainPopulationCohortId cohort start date to begin including visits.

`modelPopulationCohortIdEndDay`
The number of days relative to the mainPopulationCohortId cohort start date to end including visits.

`inclusionEvaluationCohortId`
The number of the cohort of the population to be used to designate which visits are eligible to be in the evaluation cohort

`inclusionEvaluationDaysFromStart`
The number of days from the cohort start date of the inclusionEvaluationCohortId to start eligible included visits

`inclusionEvaluationDaysFromEnd`
The number of days from the cohort start date of the inclusionEvaluationCohortId to end eligible included visits

`duringInclusionEvaluationOnly`
Only include visits that are within the cohort start and end dates

`exclusionEvaluationCohortId`
The number of the cohort of the population to be used to designate which visits are NOT eligible to be in the evaluation cohort

`exclusionEvaluationDaysFromStart`
The number of days from the cohort start date of the exclusionEvaluationCohortId to start ineligible included visits

`exclusionEvaluationDaysFromEnd`
The number of days from the cohort start date of the exclusionEvaluationCohortId to end ineligible included visits

`priorModelToUse`
folder where a previously developed model to use in analysis will be found

`minimumOffsetFromStart`
Minimum number of days to offset for the analysis visit from the start of the observation period

`minimumOffsetFromEnd`
Minimum number of days to offset for the analysis visit from the end of the observation period

`modelBaseSampleSize`
: The number of non-xSpec subjects to include in the model

`baseSampleSize`
: The maximum number of subjects in the evaluation cohort.

`lowerAgeLimit`
: The lower age for subjects in the model.

`upperAgeLimit`
: The upper age for subjects in the model.

`visitLength`
: The minimum length of index visit for acute outcomes.

`visitType`
: The concept_id for the visit type.

`gender`
: The gender(s) to be included.

`race`
: The race(s) to be included.

`ethnicity`
: The ethnicity(s) to be included.

`startDate`
: The starting date for including subjects in the model.

`endDate`
: The ending date for including subjects in the model.

`falsePositiveNegativeSubjects`
: Number of subjects to include for evaluating false positives and negatives

`cdmVersion`
: The CDM version of the database.

`outFolder`
: The folder where the output files will be written.

`exportFolder`
: The folder where the csv output files will be written.

`modelId`
: A string used to generate the file names for this model.

`evaluationCohortId`
: A string used to generate the file names for this evaluation cohort.

`excludeModelFromEvaluation`
: Should subjects used in the model be excluded from the evaluation cohort?

`randomVisitTable`
: Table stored in work directory with pre-selected random visits in format of visit_occurrence table

`removeSubjectsWithFutureDates`
: For buggy data with data in the future: ignore subjects with dates in the future?

`saveEvaluationCohortPlpData`
: Should the large PLP file for the evaluation cohort be saved? To be used for debugging purposes.

Details

Fits a diagnostic prediction model, and uses it to create an evaluation cohort with probabilities for the health outcome of interest.

Value

No return value, called for side effects. Creates database tables and output files containing the predictive model and evaluation cohort data with predicted probabilities for the health outcome.

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(
  dbms = "postgresql",
  server = "myserver",
  user = "user",
  password = "password"
)
createEvaluationCohort(
  connectionDetails = connectionDetails,
  phenotype = "Diabetes",
  analysisName = "Main",
  runDateTime = format(Sys.time(), "%b %d %Y %X"),
  databaseId = "myDB",
  xSpecCohortId = 1,
  xSensCohortId = 2,
  prevalenceCohortId = 3,
  caseCohortId = 3,
  caseFirstOccurrenceOnly = TRUE,
  cdmDatabaseSchema = "cdm.dbo",
  cohortDatabaseSchema = "results.dbo",
  cohortTable = "cohort",
  workDatabaseSchema = "scratch.dbo",
  outFolder = tempdir(),
  exportFolder = file.path(tempdir(), "export")
)

## End(Not run)
```

`createPheValuatorAnalysis` *Create a PheValuator analysis specification*

Description

Create a PheValuator analysis specification

Usage

```
createPheValuatorAnalysis(
  analysisId,
  description,
  createEvaluationCohortArgs,
  testPhenotypeAlgorithmArgs
)
```

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.
createEvaluationCohortArgs	An object representing the arguments to be used when calling the createCreateEvaluationCohortArgs function.
testPhenotypeAlgorithmArgs	Should the createTestPhenotypeAlgorithmArgs function be used in this analysis?

Details

Create a set of analysis choices, to be used with the runPheValuatorAnalyses function.

Value

A list object of class "pheValuatorAnalysis" containing the analysis specification. Includes the analysis ID, description, and argument specifications for creating the evaluation cohort and testing the phenotype algorithm.

Examples

```
evalCohortArgs <- createCreateEvaluationCohortArgs(
  xSpecCohortId = 1,
  xSensCohortId = 2,
  prevalenceCohortId = 3
)
testArgs <- createTestPhenotypeAlgorithmArgs(
  phenotypeCohortId = 10
)
analysis <- createPheValuatorAnalysis(
  analysisId = 1,
  description = "Main analysis",
  createEvaluationCohortArgs = evalCohortArgs,
  testPhenotypeAlgorithmArgs = testArgs
)
```

---

createTestPhenotypeAlgorithmArgs

Create a parameter object for the function testPhenotypeAlgorithm

---

Description

Create a parameter object for the function testPhenotypeAlgorithm

Usage

```
createTestPhenotypeAlgorithmArgs(
  cutPoints = c("EV"),
  phenotypeCohortId,
  washoutPeriod = 0,
  splayPrior = 7,
  splayPost = 7
)
```

Arguments

cutPoints	A list of threshold predictions for the evaluations. Include "EV" for the expected value
phenotypeCohortId	The ID of the cohort to evaluate in the specified cohort table.
washoutPeriod	The mininum required continuous observation time prior to index date for subjects within the cohort to test (Default = 0).
splayPrior	The number of days to allow for test phenotype visit date prior to evaluation date
splayPost	The number of days to allow for test phenotype visit date after evaluation date

Details

Create an object defining the parameter values.

Value

A list object of class "args" containing all the provided parameters. This object is used as an argument to the testPhenotypeAlgorithm function.

Examples

```
testArgs <- createTestPhenotypeAlgorithmArgs(
  phenotypeCohortId = 10,
  cutPoints = c("EV", "0.5")
)
```

---

loadPheValuatorAnalysisList

Load a list of phevaluatorAnalysis from file

Description

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

Usage

```
loadPheValuatorAnalysisList(file)
```

Arguments

file	The name of the file

Value

A list of objects of type pheValuatorAnalysis.

Examples

```
## Not run:
analysisList <- loadPheValuatorAnalysisList(file = "analysisList.json")

## End(Not run)
```

---

runPheValuatorAnalyses

Run a list of analyses

Description

Run a list of analyses

Usage

```
runPheValuatorAnalyses(
  phenotype,
  cohortDefinitionSet = data.frame(),
  analysisName = "Main",
  runDateTime = format(Sys.time(), "%b %d %Y %X"),
  connectionDetails,
  oracleTempSchema = NULL,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  cdmDatabaseSchema,
  cohortDatabaseSchema = cdmDatabaseSchema,
  cohortTable = "cohort",
  workDatabaseSchema = cdmDatabaseSchema,
  databaseId = cdmDatabaseSchema,
  cdmVersion = 5,
  outputFolder,
  priorModelToUse = NULL,
  pheValuatorAnalysisList
)
```

Arguments

phenotype	Name of the phenotype for analysis
cohortDefinitionSet	Data.frame of cohorts must include columns cohortId, cohortName, json, sql - should include all cohort definitions needed to replicate PheValuator analysis
analysisName	Name of the analysis
runDateTime	Starting date and time of the PheValuator run
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	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'.
cohortDatabaseSchema	The name of the database schema that is the location where the cohort data used to define the at risk cohort is available. Requires read permissions to this database.
cohortTable	The tablename that contains the at risk cohort. The expectation is cohortTable has format of COHORT table: cohort_concept_id, SUBJECT_ID, COHORT_START_DATE, COHORT_END_DATE.
workDatabaseSchema	The name of the database schema that is the location where a table can be created and afterwards removed. Requires write permissions to this database.
databaseId	Name of the database in the analysis
cdmVersion	Define the OMOP CDM version used: currently supports "5".
outputFolder	Name of the folder where all the outputs will be written to.
priorModelToUse	folder where a previously developed model to use in analysis will be found
pheValuatorAnalysisList	A list of objects of type pheValuatorAnalysis as created using the createPheValuatorAnalysis function.

Details

Run a list of analyses.

Value

A data frame specifying where the constructed evaluation cohort and phenotype evaluation results can be found in the local file system.

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(
  dbms = "postgresql",
  server = "myserver",
  user = "user",
  password = "password"
)
analysisList <- list(
  createPheValuatorAnalysis(
    analysisId = 1,
    description = "Main analysis",
    createEvaluationCohortArgs = createCreateEvaluationCohortArgs(
      xSpecCohortId = 1,
      prevalenceCohortId = 3
    ),
    testPhenotypeAlgorithmArgs = createTestPhenotypeAlgorithmArgs(
      phenotypeCohortId = 10
    )
  )
)
runPheValuatorAnalyses(
  phenotype = "Diabetes",
  connectionDetails = connectionDetails,
  cdmDatabaseSchema = "cdm.dbo",
  outputFolder = tempdir(),
  pheValuatorAnalysisList = analysisList
)

## End(Not run)
```

---

savePheValuatorAnalysisList

Save a list of phevaluatorAnalysis to file

Description

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

Usage

```
savePheValuatorAnalysisList(pheValuatorAnalysisList, file)
```

Arguments

pheValuatorAnalysisList	The pheValuatorAnalysis list to be written to file
file	The name of the file where the results will be written

Value

No return value, called for side effects. Writes a JSON file containing the analysis list to the specified file path.

Examples

```
## Not run:
# Requires pre-existing analysis list object; example usage:
# savePheValuatorAnalysisList(pheValuatorAnalysisList = myAnalysisList,
#                             file = file.path(tempdir(), "analysisList.json"))

## End(Not run)
```

---

summarizePheValuatorAnalyses

Summarize results of PheValuator analyses

Description

Summarize results of PheValuator analyses

Usage

```
summarizePheValuatorAnalyses(referenceTable, outputFolder)
```

Arguments

referenceTable	A reference table as created using runPheValuatorAnalyses.
outputFolder	The output folder used when calling runPheValuatorAnalyses.

Value

A data frame of results.

Examples

```
## Not run:
refTable <- runPheValuatorAnalyses(
  phenotype = "Diabetes",
  connectionDetails = connectionDetails,
  cdmDatabaseSchema = "cdm.dbo",
  outputFolder = tempdir(),
  pheValuatorAnalysisList = analysisList
)
results <- summarizePheValuatorAnalyses(refTable, outputFolder = tempdir())

## End(Not run)
```

---

testPhenotypeAlgorithm

Test phenotype algorithms

Description

Test phenotype algorithms

Usage

```
testPhenotypeAlgorithm(
  phenotype,
  analysisName,
  runDateTime,
  connectionDetails,
  cutPoints = c("EV"),
  outFolder,
  exportFolder,
  evaluationCohortId = "main",
  phenotypeCohortId,
  cdmDatabaseSchema,
  databaseId,
  cohortDatabaseSchema,
  cohortTable,
  washoutPeriod = 0,
  splayPrior = 7,
  splayPost = 7
)
```

Arguments

phenotype	Name of the phenotype for analysis
analysisName	Name of the analysis

`runDateTime` Starting date and time of the PheValuator run

`connectionDetails`
ConnectionDetails created using the function createConnectionDetails in the DatabaseConnector package.

`cutPoints` A list of threshold predictions for the evaluations. Include "EV" for the expected value

`outFolder` The folder where the cohort evaluation output files are written

`exportFolder` The folder where the csv output files will be written.

`evaluationCohortId`
A string used to generate the file names for the evaluation cohort.

`phenotypeCohortId`
The ID of the cohort to evaluate in the specified cohort table.

`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'.

`databaseId` Name of the database in the analysis

`cohortDatabaseSchema`
The name of the database schema that is the location where the cohort data used to define the at risk cohort is available. Requires read permissions to this database.

`cohortTable` The tablename that contains the at risk cohort. The expectation is cohortTable has format of COHORT table: cohort_concept_id, SUBJECT_ID, COHORT_START_DATE, COHORT_END_DATE.

`washoutPeriod` The mininum required continuous observation time prior to index date for subjects within the cohort to test (Default = 0).

`splayPrior` The number of days to allow for test phenotype visit date prior to evaluation date

`splayPost` The number of days to allow for test phenotype visit date after evaluation date

Details

This function will perform the phenotype algorithm evaluation using the evaluation cohort returned from createEvalCohort and the phenotype algorithm cohort specified

Value

A dataframe with the results from the phenotype algorithm evaluation.

If 0.5 is included as a cutpoint, the data frame will have an attribute called 'misses', a dataframe with a sample of subject ids for TPs, FPs, TNs, and FNs for the 50 percent and over prediction threshold.

Examples

```
## Not run:
connectionDetails <- DatabaseConnector::createConnectionDetails(
  dbms = "postgresql",
  server = "myserver",
  user = "user",
  password = "password"
)
results <- testPhenotypeAlgorithm(
  phenotype = "Diabetes",
  analysisName = "Main",
  runDateTime = format(Sys.time(), "%b %d %Y %X"),
  connectionDetails = connectionDetails,
  cutPoints = c("EV", "0.5"),
  outFolder = tempdir(),
  exportFolder = file.path(tempdir(), "export"),
  phenotypeCohortId = 10,
  cdmDatabaseSchema = "cdm.dbo",
  databaseId = "myDB",
  cohortDatabaseSchema = "results.dbo",
  cohortTable = "cohort"
)

## End(Not run)
```

Index

computePerformanceMetricsFromCounts, 2
createCreateEvaluationCohortArgs, 3, *13*
createDefaultCovariateSettings, 6
createEvaluationCohort, *6*, 7
createPheValuatorAnalysis, 12, *16*
createTestPhenotypeAlgorithmArgs, *13*, 13

loadPheValuatorAnalysisList, 14

runPheValuatorAnalyses, *13*, 15, *18*

savePheValuatorAnalysisList, 17
summarizePheValuatorAnalyses, 18

testPhenotypeAlgorithm, *2*, *14*, 19