

Package ‘SelfControlledCohort’

January 6, 2022

Type Package

Title Population-Level Estimation Method that Estimates Incidence Rate Comparison of Exposed/Unexposed Time Within an Exposed Cohort

Version 1.6.0

Date 2022-01-05

Maintainer Jamie Gilbert <jpg@azimov.co.uk>

Description A method to estimate risk by comparing time exposed with time unexposed among the exposed cohort.

URL <https://github.com/OHDSI/SelfControlledCohort>

BugReports <https://github.com/OHDSI/SelfControlledCohort/issues>

Depends DatabaseConnector (≥ 5.0.0),
R (≥ 4.0.0)

Imports SqlRender (≥ 1.4.3),
ParallelLogger,
rateratio.test

Suggests withr,
testthat,
knitr,
rmarkdown,
Eunomia,
DiagrammeR,
dplyr,
tidyr,
R.utils

Remotes ohdsi/Eunomia

License Apache License 2.0

RoxygenNote 7.1.2

VignetteBuilder knitr

R topics documented:

createExposureOutcome	2
createRunSelfControlledCohortArgs	2
createScAnalysis	4

getSccRiskWindowStats	5
loadExposureOutcomeList	6
loadSccAnalysisList	7
runSccAnalyses	7
runSccRiskWindows	9
runSelfControlledCohort	11
saveExposureOutcomeList	14
saveSccAnalysisList	15
summarizeAnalyses	15

`createExposureOutcome` *Create exposure-outcome combinations.*

Description

Create exposure-outcome combinations.

Usage

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

Arguments

<code>exposureId</code>	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 <code>exposureType</code> parameter in the createSccAnalysis function.
<code>outcomeId</code>	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 <code>#'</code> <code>outcomeType</code> parameter in the createSccAnalysis function.

Details

Create a hypothesis of interest, to be used with the [runSccAnalyses](#) function.

`createRunSelfControlledCohortArgs`

Create a parameter object for the function `runSelfControlledCohort`

Description

Create a parameter object for the function `runSelfControlledCohort`

Usage

```

createRunSelfControlledCohortArgs(
  firstExposureOnly = TRUE,
  firstOutcomeOnly = TRUE,
  minAge = "",
  maxAge = "",
  studyStartDate = "",
  studyEndDate = "",
  addLengthOfExposureExposed = TRUE,
  riskWindowStartExposed = 1,
  riskWindowEndExposed = 30,
  addLengthOfExposureUnexposed = TRUE,
  riskWindowEndUnexposed = -1,
  riskWindowStartUnexposed = -30,
  hasFullTimeAtRisk = FALSE,
  washoutPeriod = 0,
  followupPeriod = 0,
  computeTarDistribution = FALSE
)

```

Arguments

firstExposureOnly	If TRUE, only use first occurrence of each drug concept id for each person
firstOutcomeOnly	If TRUE, only use first occurrence of each condition conceptid for each person.
minAge	Integer for minimum allowable age.
maxAge	Integer for maximum allowable age.
studyStartDate	Date for minimum allowable data for index exposure. Dateformat is 'yyyymmdd'.
studyEndDate	Date for maximum allowable data for index exposure. Dateformat is 'yyyymmdd'.
addLengthOfExposureExposed	If TRUE, use the duration from drugEraStart -i drugEraEnd as part of timeAtRisk.
riskWindowStartExposed	Integer of days to add to drugEraStart for start of timeAtRisk (0 to include index date, 1 to start the dayafter).
riskWindowEndExposed	Additional window to add to end of exposure period (if addLengthOfExposureExposed = TRUE, then add to exposure enddate, else add to exposure start date).
addLengthOfExposureUnexposed	If TRUE, use the duration from exposure start -i exposureend as part of timeAtRisk looking back before exposurestart.
riskWindowEndUnexposed	Integer of days to add to exposure start for end of timeAtRisk (0 to include index date, -1 to end the daybefore).

riskWindowStartUnexposed Additional window to add to start of exposure period (if `addLengthOfExposureUnexposed = TRUE`, then add to exposure end date, else add to exposure start date).

hasFullTimeAtRisk If `TRUE`, restrict to people who have full time-at-risk exposed and unexposed.

washoutPeriod Integer to define required time observed before exposure start.

followupPeriod Integer to define required time observed after exposure start.

computeTarDistribution If `TRUE`, compute the distribution of time-at-risk and average absolute time between treatment and outcome. Note, may add significant computation time on some database engines. If set `true` in one analysis will default to `true` for all others.

Details

Create an object defining the parameter values.

<code>createSccAnalysis</code>	<i>Create a SelfControlledCohort analysis specification</i>
--------------------------------	---

Description

Create a `SelfControlledCohort` analysis specification

Usage

```
createSccAnalysis(
  analysisId = 1,
  description = "",
  exposureType = NULL,
  outcomeType = NULL,
  runSelfControlledCohortArgs
)
```

Arguments

analysisId An integer that will be used later to refer to this specific set of analysis choices.

description A short description of the analysis.

exposureType If more than one exposure is provided for each `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.

runSelfControlledCohortArgs An object representing the arguments to be used when calling the [runSelfControlledCohort](#) function.

Details

Create a set of analysis choices, to be used with the [runSccAnalyses](#) function.

`getSccRiskWindowStats` *Get Self-Controlled Cohort Risk Window Statistics*

Description

Compute statistics from risk windows.

Usage

```
getSccRiskWindowStats(
  connection,
  outcomeDatabaseSchema,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  oracleTempSchema = NULL,
  outcomeIds = NULL,
  cdmVersion = 5,
  outcomeTable = "condition_era",
  firstOutcomeOnly = TRUE,
  resultsDatabaseSchema = NULL,
  riskWindowsTable = "#risk_windows"
)
```

Arguments

<code>connection</code>	DatabaseConnector connection instance
<code>outcomeDatabaseSchema</code>	The name of the database schema that is the location where the data used to define the outcome cohorts is available. If <code>exposureTable = CONDITION_ERA</code> , <code>exposureDatabaseSchema</code> is not used by assumed to be <code>cdmSchema</code> . Requires read permissions to this database.
<code>tempEmulationSchema</code>	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.
<code>oracleTempSchema</code>	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.
<code>outcomeIds</code>	The <code>condition_concept_ids</code> or <code>cohort_definition_ids</code> of the outcomes of interest. If empty, all the outcomes in the outcome table will be included.
<code>cdmVersion</code>	Define the OMOP CDM version used: currently support "4" and "5".
<code>outcomeTable</code>	The tablename that contains the outcome cohorts. If <code>outcomeTable != CONDITION_OCCURRENCE</code> , then expectation is <code>outcomeTable</code> has format of COHORT table: <code>COHORT_DEFINITION_ID</code> , <code>SUBJECT_ID</code> , <code>COHORT_START_DATE</code> , <code>COHORT_END_DATE</code> .
<code>firstOutcomeOnly</code>	If TRUE, only use first occurrence of each condition concept id for each person.

resultsDatabaseSchema
 Schema to output results to. Ignored if resultsTable and riskWindowsTable are temporary.

riskWindowsTable
 String: optionally store the risk windows in a (non-temporary) table.

Details

Requires a risk window table to be created first with ‘runSccRiskWindows’

Value

list containing data frames: treatmentTimeDistribution, timeToOutcomeDistribution, timeToOutcomeDistributionExposed, timeToOutcomeDistributionUnexposed

Examples

```
## Not run:
# First, create the risk windows table
connectionDetails <- Eunomia::getEunomiaConnectionDetails()
connection <- DatabaseConnector::connect(connectionDetails)
riskWindowsTable <- "computed_risk_windows"
runSccRiskWindows(connection,
  cdmDatabaseSchema = "main",
  exposureIds = c(1102527, 1125315),
  resultsDatabaseSchema = "main", # This is the schema where the results will be stored
  riskWindowsTable = riskWindowsTable,
  exposureTable = "drug_era")
# Get stats based on outcomes of interest
tarStats <- getSccRiskWindowStats(connection,
  outcomeDatabaseSchema = "main",
  resultsDatabaseSchema = "main",
  riskWindowsTable = riskWindowsTable,
  outcomeTable = "condition_era",
  outcomeIds = 192671)

## End(Not run)
```

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`.

<code>loadSccAnalysisList</code>	<i>Load a list of sccAnalysis from file</i>
----------------------------------	---

Description

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

Usage

```
loadSccAnalysisList(file)
```

Arguments

<code>file</code>	The name of the file
-------------------	----------------------

Value

A list of objects of type `sccAnalysis`.

<code>runSccAnalyses</code>	<i>Run a list of analyses</i>
-----------------------------	-------------------------------

Description

Run a list of analyses

Usage

```
runSccAnalyses(  
  connectionDetails,  
  cdmDatabaseSchema,  
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),  
  oracleTempSchema = NULL,  
  exposureDatabaseSchema = cdmDatabaseSchema,  
  exposureTable = "drug_era",  
  outcomeDatabaseSchema = cdmDatabaseSchema,  
  outcomeTable = "condition_occurrence",  
  cdmVersion = 5,  
  outputFolder = "./SelfControlledCohortOutput",  
  sccAnalysisList,  
  exposureOutcomeList,  
  analysisThreads = 1,  
  computeThreads = 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 the OMOP CDM and vocabulary.

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.

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_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 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.

sccAnalysisList

A list of objects of type `sccAnalysis` as created using the `createSccAnalysis` function.

exposureOutcomeList

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

analysisThreads

The number of parallel threads to use to execute the analyses.

computeThreads

Number of parallel threads for computing IRRs with exact confidence intervals.

Details

Run a list of analyses for the drug-comparator-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(cmAnalysisList) * length(drugComparatorOutcomesList)'`.

runSccRiskWindows	<i>Run Self-Controlled Cohort Risk Windows</i>
-------------------	--

Description

Compute time at risk exposed and time at risk unexposed for risk window parameters. See ‘getSccRiskWindowStats’ for example usage.

Usage

```
runSccRiskWindows(
  connection,
  cdmDatabaseSchema,
  cdmVersion = 5,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  oracleTempSchema = NULL,
  exposureIds = NULL,
  exposureDatabaseSchema = cdmDatabaseSchema,
  exposureTable = "drug_era",
  firstExposureOnly = TRUE,
  minAge = "",
  maxAge = "",
  studyStartDate = "",
  studyEndDate = "",
  addLengthOfExposureExposed = TRUE,
  riskWindowStartExposed = 1,
  riskWindowEndExposed = 30,
  addLengthOfExposureUnexposed = TRUE,
  riskWindowEndUnexposed = -1,
  riskWindowStartUnexposed = -30,
  hasFullTimeAtRisk = FALSE,
  washoutPeriod = 0,
  followupPeriod = 0,
  riskWindowsTable = "#risk_windows",
  resultsDatabaseSchema = NULL
)
```

Arguments

connection	DatabaseConnector connection instance
cdmDatabaseSchema	Name of database schema that contains the OMOP CDM and vocabulary.
cdmVersion	Define the OMOP CDM version used: currently support "4" and "5".
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.
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.

<code>exposureIds</code>	A vector containing the <code>drug_concept_ids</code> or <code>cohort_definition_ids</code> of the exposures of interest. If empty, all exposures in the exposure table will be included.
<code>exposureDatabaseSchema</code>	The name of the database schema that is the location where the exposure data used to define the exposure cohorts is available. If <code>exposureTable = DRUG_ERA</code> , <code>exposureDatabaseSchema</code> is not used by assumed to be <code>cdmSchema</code> . Requires read permissions to this database.
<code>exposureTable</code>	The tablename that contains the exposure cohorts. If <code>exposureTable != DRUG_ERA</code> , then expectation is <code>exposureTable</code> has format of COHORT table: <code>cohort_concept_id</code> , <code>SUBJECT_ID</code> , <code>COHORT_START_DATE</code> , <code>COHORT_END_DATE</code> .
<code>firstExposureOnly</code>	If TRUE, only use first occurrence of each drug concept id for each person
<code>minAge</code>	Integer for minimum allowable age.
<code>maxAge</code>	Integer for maximum allowable age.
<code>studyStartDate</code>	Date for minimum allowable data for index exposure. Date format is 'yyyymmdd'.
<code>studyEndDate</code>	Date for maximum allowable data for index exposure. Date format is 'yyyymmdd'.
<code>addLengthOfExposureExposed</code>	If TRUE, use the duration from <code>drugEraStart</code> -i- <code>drugEraEnd</code> as part of <code>timeAtRisk</code> .
<code>riskWindowStartExposed</code>	Integer of days to add to <code>drugEraStart</code> for start of <code>timeAtRisk</code> (0 to include index date, 1 to start the day after).
<code>riskWindowEndExposed</code>	Additional window to add to end of exposure period (if <code>addLengthOfExposureExposed = TRUE</code> , then add to exposure end date, else add to exposure start date).
<code>addLengthOfExposureUnexposed</code>	If TRUE, use the duration from exposure start -i- exposure end as part of <code>timeAtRisk</code> looking back before exposure start.
<code>riskWindowEndUnexposed</code>	Integer of days to add to exposure start for end of <code>timeAtRisk</code> (0 to include index date, -1 to end the day before).
<code>riskWindowStartUnexposed</code>	Additional window to add to start of exposure period (if <code>addLengthOfExposureUnexposed = TRUE</code> , then add to exposure end date, else add to exposure start date).
<code>hasFullTimeAtRisk</code>	If TRUE, restrict to people who have full time-at-risk exposed and unexposed.
<code>washoutPeriod</code>	Integer to define required time observed before exposure start.
<code>followupPeriod</code>	Integer to define required time observed after exposure start.
<code>riskWindowsTable</code>	String: optionally store the risk windows in a (non-temporary) table.
<code>resultsDatabaseSchema</code>	Schema to output results to. Ignored if <code>resultsTable</code> and <code>riskWindowsTable</code> are temporary.

runSelfControlledCohort

Run self-controlled cohort

Description

runSelfControlledCohort generates population-level estimation by comparing exposed and unexposed time among exposed cohort.

Usage

```
runSelfControlledCohort(
  connectionDetails = NULL,
  cdmDatabaseSchema,
  connection = NULL,
  cdmVersion = 5,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  oracleTempSchema = NULL,
  exposureIds = NULL,
  outcomeIds = NULL,
  exposureDatabaseSchema = cdmDatabaseSchema,
  exposureTable = "drug_era",
  outcomeDatabaseSchema = cdmDatabaseSchema,
  outcomeTable = "condition_era",
  firstExposureOnly = TRUE,
  firstOutcomeOnly = TRUE,
  minAge = "",
  maxAge = "",
  studyStartDate = "",
  studyEndDate = "",
  addLengthOfExposureExposed = TRUE,
  riskWindowStartExposed = 1,
  riskWindowEndExposed = 30,
  addLengthOfExposureUnexposed = TRUE,
  riskWindowEndUnexposed = -1,
  riskWindowStartUnexposed = -30,
  hasFullTimeAtRisk = FALSE,
  washoutPeriod = 0,
  followupPeriod = 0,
  computeTarDistribution = FALSE,
  computeThreads = 1,
  riskWindowsTable = "#risk_windows",
  resultsTable = "#results",
  resultsDatabaseSchema = NULL,
  postProcessFunction = NULL,
  postProcessArgs = list(),
  returnEstimates = TRUE
)
```

Arguments**connectionDetails**

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

cdmDatabaseSchema

Name of database schema that contains the OMOP CDM and vocabulary.

connection DatabaseConnector connection instance**cdmVersion** Define the OMOP CDM version used: currently support "4" and "5".**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.

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.

exposureIds A vector containing the `drug_concept_ids` or `cohort_definition_ids` of the exposures of interest. If empty, all exposures in the exposure table will be included.**outcomeIds** The `condition_concept_ids` or `cohort_definition_ids` of the outcomes of interest. If empty, all the outcomes in the outcome table will be included.**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_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 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`.**firstExposureOnly**

If TRUE, only use first occurrence of each drug concept id for each person

firstOutcomeOnly

If TRUE, only use first occurrence of each condition concept id for each person.

minAge Integer for minimum allowable age.**maxAge** Integer for maximum allowable age.**studyStartDate** Date for minimum allowable data for index exposure. Date format is 'yyyymmdd'.

studyEndDate	Date for maximum allowable data for index exposure. Date format is 'yyyymmdd'.
addLengthOfExposureExposed	If TRUE, use the duration from drugEraStart -j drugEraEnd as part of timeAtRisk.
riskWindowStartExposed	Integer of days to add to drugEraStart for start of timeAtRisk (0 to include index date, 1 to start the day after).
riskWindowEndExposed	Additional window to add to end of exposure period (if addLengthOfExposureExposed = TRUE, then add to exposure end date, else add to exposure start date).
addLengthOfExposureUnexposed	If TRUE, use the duration from exposure start -j exposure end as part of timeAtRisk looking back before exposure start.
riskWindowEndUnexposed	Integer of days to add to exposure start for end of timeAtRisk (0 to include index date, -1 to end the day before).
riskWindowStartUnexposed	Additional window to add to start of exposure period (if addLengthOfExposureUnexposed = TRUE, then add to exposure end date, else add to exposure start date).
hasFullTimeAtRisk	If TRUE, restrict to people who have full time-at-risk exposed and unexposed.
washoutPeriod	Integer to define required time observed before exposure start.
followupPeriod	Integer to define required time observed after exposure start.
computeTarDistribution	If TRUE, computer the distribution of time-at-risk and average absolute time between treatment and outcome. Note, may add significant computation time on some database engines.
computeThreads	Number of parallel threads for computing IRRs with exact confidence intervals.
riskWindowsTable	String: optionally store the risk windows in a (non-temporary) table.
resultsTable	String: optionally store the summary results (number exposed/ unexposed patients per outcome-exposure pair) in a (non-temporary) table. Note that this table does not store the rate ratios, only the values required to calculate rate ratios.
resultsDatabaseSchema	Schema to oputput results to. Ignored if resultsTable and riskWindowsTable are temporary.
postProcessFunction	Callback function to handle batches of data. Useful for massive result sets that overflow system memory. See example.
postProcessArgs	Arguments for post processing function callback.
returnEstimates	Boolean opt to not return estimates, only useful in the case where post-ProcessFunction is used

Details

Population-level estimation method that estimates incidence rate comparison of exposed/unexposed time within an exposed cohort. If multiple exposureIds and outcomeIds are provided, estimates will be generated for every combination of exposure and outcome.

Value

An object of type `sccResults` containing the results of the analysis.

References

Ryan PB, Schuemie MJ, Madigan D. Empirical performance of a self-controlled cohort method: lessons for developing a risk identification and analysis system. *Drug Safety* 36 Suppl1:S95-106, 2013

Examples

```
## Not run:
connectionDetails <- createConnectionDetails(dbms = "sql server",
                                             server = "RNDUSRDHIT07.jnj.com")
sccResult <- runSelfControlledCohort(connectionDetails,
                                     cdmDatabaseSchema = "cdm_truven_mdcrr.dbo",
                                     exposureIds = c(767410, 1314924, 907879),
                                     outcomeIds = 444382,
                                     outcomeTable = "condition_era")

# Using a callback function that writes data to a csv file and not store in memory
csvFileName <- "D:/path/to/output.csv"
writeSccData <- function(data, position, csvFileName) {
  vroom::vroom_write(data, csvFileName, delim = ",", append = position != 1, na = "")
}

runSelfControlledCohort(connectionDetails,
                        cdmDatabaseSchema = "cdm_truven_mdcrr.dbo",
                        exposureIds = c(767410, 1314924, 907879),
                        outcomeIds = 444382,
                        outcomeTable = "condition_era",
                        postProcessFunction = writeSccData,
                        postProcessArgs = list(csvFileName = csvFileName),
                        returnEstimates = FALSE)

## End(Not run)
```

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

saveSccAnalysisList	<i>Save a list of sccAnalysis to file</i>
---------------------	---

Description

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

Usage

```
saveSccAnalysisList(sccAnalysisList, file)
```

Arguments

sccAnalysisList	The sccAnalysis list to be written to file
file	The name of the file where the results will be written

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

Description

Create a summary report of the analyses

Usage

```
summarizeAnalyses(resultsReference, outputFolder)
```

Arguments

resultsReference	A data.frame as created by the runSccAnalyses function.
outputFolder	Name of the folder where all the outputs have been written to.