

# Package ‘CohortGenerator’

December 6, 2021

**Type** Package

**Title** An R Package for Cohort Generation Against the OMOP CDM

**Version** 0.2.0

**Date** 2021-11-17

**Maintainer** Anthony Sena <sena@ohdsi.org>

**Description**

An R package for that encapsulates the functions for generating cohorts against the OMOP CDM.

**Depends** DatabaseConnector (>= 5.0.0),  
R (>= 3.6.0)

**Imports** checkmate,  
digest,  
dplyr,  
lubridate,  
ParallelLogger (>= 2.0.2),  
readr (>= 1.4.0),  
rlang,  
RJSONIO,  
SqlRender (>= 1.7.0)

**Suggests** CirceR (>= 1.1.1),  
Eunomia,  
knitr,  
rmarkdown,  
ROhdsiWebApi,  
testthat

**Remotes** ohdsi/CirceR,  
ohdsi/Eunomia,  
ohdsi/ROhdsiWebApi

**License** Apache License

**VignetteBuilder** knitr

**URL** <https://ohdsi.github.io/CohortGenerator/>, <https://github.com/OHDSI/CohortGenerator>

**BugReports** <https://github.com/OHDSI/CohortGenerator/issues>

**RoxygenNote** 7.1.1

**Encoding** UTF-8

**Language** en-US

R topics documented:

computeChecksum . . . . .	2
createCohortTables . . . . .	2
createEmptyCohortDefinitionSet . . . . .	3
dropCohortStatsTables . . . . .	4
exportCohortStatsTables . . . . .	4
generateCohort . . . . .	5
generateCohortSet . . . . .	6
getCohortCounts . . . . .	8
getCohortTableNames . . . . .	9
getRequiredTasks . . . . .	10
insertInclusionRuleNames . . . . .	10
isTaskRequired . . . . .	11
recordTasksDone . . . . .	12
saveIncremental . . . . .	12
<b>Index</b>	<b>13</b>

---

computeChecksum	<i>Computes the checksum for a value</i>
-----------------	--

---

Description

This is used as part of the incremental operations to hash a value to store in a record keeping file. This function leverages the md5 hash from the digest package

Usage

computeChecksum(val)

Arguments

val                      The value to hash. It is converted to a character to perform the hash.

Value

Returns a string containing the checksum

---

createCohortTables	<i>Create cohort tables</i>
--------------------	-----------------------------

---

Description

This function creates an empty cohort table and empty tables for cohort statistics.

## Usage

```
createCohortTables(  
  connectionDetails = NULL,  
  connection = NULL,  
  cohortDatabaseSchema,  
  cohortTableNames = getCohortTableNames(),  
  incremental = FALSE  
)
```

## Arguments

connectionDetails	An object of type connectionDetails as created using the <a href="#">createConnectionDetails</a> function in the DatabaseConnector package. Can be left NULL if connection is provided.
connection	An object of type connection as created using the <a href="#">connect</a> function in the DatabaseConnector package. Can be left NULL if connectionDetails is provided, in which case a new connection will be opened at the start of the function, and closed when the function finishes.
cohortDatabaseSchema	The schema to hold the cohort tables. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
cohortTableNames	The names of the cohort tables. See <a href="#">getCohortTableNames</a> for more details.
incremental	When set to TRUE, this function will check to see if the cohortTableNames exists in the cohortDatabaseSchema and if they exist, it will skip creating the tables.

---

createEmptyCohortDefinitionSet

*Create an empty cohort definition set*

---

## Description

This function creates an empty cohort set data.frame for use with generateCohortSet.

## Usage

```
createEmptyCohortDefinitionSet()
```

## Value

Returns an empty cohort set data.frame

---

dropCohortStatsTables    *Drop cohort statistics tables*

---

### Description

This function drops the cohort statistics tables.

### Usage

```
dropCohortStatsTables(
  connectionDetails = NULL,
  connection = NULL,
  cohortDatabaseSchema,
  cohortTableNames = getCohortTableNames()
)
```

### Arguments

connectionDetails	An object of type connectionDetails as created using the <a href="#">createConnectionDetails</a> function in the DatabaseConnector package. Can be left NULL if connection is provided.
connection	An object of type connection as created using the <a href="#">connect</a> function in the DatabaseConnector package. Can be left NULL if connectionDetails is provided, in which case a new connection will be opened at the start of the function, and closed when the function finishes.
cohortDatabaseSchema	The schema to hold the cohort tables. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
cohortTableNames	The names of the cohort statistics tables. See <a href="#">getCohortTableNames</a> for more details.

---

exportCohortStatsTables

*Export the cohort statistics tables to the file system*

---

### Description

This function retrieves the data from the cohort statistics tables and writes them to the inclusion statistics folder specified in the function call.

### Usage

```
exportCohortStatsTables(
  connectionDetails,
  connection = NULL,
  cohortDatabaseSchema,
  cohortTableNames = getCohortTableNames(),
```

```

        cohortStatisticsFolder,
        incremental
    )

```

## Arguments

connectionDetails	An object of type connectionDetails as created using the <a href="#">createConnectionDetails</a> function in the DatabaseConnector package. Can be left NULL if connection is provided.
connection	An object of type connection as created using the <a href="#">connect</a> function in the DatabaseConnector package. Can be left NULL if connectionDetails is provided, in which case a new connection will be opened at the start of the function, and closed when the function finishes.
cohortDatabaseSchema	The schema to hold the cohort tables. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
cohortTableNames	The names of the cohort statistics tables. See <a href="#">getCohortTableNames</a> for more details.
cohortStatisticsFolder	The path to the folder where the cohort statistics folder where the results will be written
incremental	If incremental = TRUE, results are written to update values instead of overwriting an existing results

---

generateCohort	<i>Generates a cohort</i>
----------------	---------------------------

---

## Description

This function is used by generateCohortSet to generate a cohort against the CDM.

## Usage

```

generateCohort(
  cohortId = NULL,
  cohortDefinitionSet,
  connection = NULL,
  connectionDetails = NULL,
  cdmDatabaseSchema,
  tempEmulationSchema,
  cohortDatabaseSchema,
  cohortTableNames,
  incremental,
  recordKeepingFile
)

```

**Arguments**

cohortId	The cohortId in the list of cohortDefinitionSet
cohortDefinitionSet	The cohortDefinitionSet argument must be a data frame with the following columns:  <b>cohortId</b> The unique integer identifier of the cohort <b>cohortName</b> The cohort's name <b>sql</b> The OHDSI-SQL used to generate the cohort Optionally, this data frame may contain: <b>json</b> The Circe JSON representation of the cohort
connection	An object of type connection as created using the <a href="#">connect</a> function in the DatabaseConnector package. Can be left NULL if connectionDetails is provided, in which case a new connection will be opened at the start of the function, and closed when the function finishes.
connectionDetails	An object of type connectionDetails as created using the <a href="#">createConnectionDetails</a> function in the DatabaseConnector package. Can be left NULL if connection is provided.
cdmDatabaseSchema	Schema name where your patient-level data in OMOP CDM format resides. Note that for SQL Server, this should include both the database and schema name, for example 'cdm_data.dbo'.
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.
cohortDatabaseSchema	Schema name where your cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
cohortTableNames	List of cohort table names.
incremental	Create only cohorts that haven't been created before?
recordKeepingFile	If incremental = TRUE, this file will contain information on cohorts already generated

---

generateCohortSet	<i>Generate a set of cohorts</i>
-------------------	----------------------------------

---

**Description**

This function generates a set of cohorts in the cohort table.

## Usage

```
generateCohortSet(
  connectionDetails = NULL,
  connection = NULL,
  cdmDatabaseSchema,
  tempEmulationSchema = getOption("sqlRenderTempEmulationSchema"),
  cohortDatabaseSchema = cdmDatabaseSchema,
  cohortTableNames = getCohortTableNames(),
  cohortDefinitionSet = NULL,
  stopOnError = TRUE,
  incremental = FALSE,
  incrementalFolder = NULL
)
```

## Arguments

- |                      |   |
|----------------------|---|
| connectionDetails    | An object of type connectionDetails as created using the <a href="#">createConnectionDetails</a> function in the DatabaseConnector package. Can be left NULL if connection is provided.   |
| connection           | An object of type connection as created using the <a href="#">connect</a> function in the DatabaseConnector package. Can be left NULL if connectionDetails is provided, in which case a new connection will be opened at the start of the function, and closed when the function finishes.  |
| cdmDatabaseSchema    | Schema name where your patient-level data in OMOP CDM format resides. Note that for SQL Server, this should include both the database and schema name, for example 'cdm_data.dbo'.  |
| 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.   |
| cohortDatabaseSchema | Schema name where your cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.  |
| cohortTableNames     | List of cohort table names.   |
| cohortDefinitionSet  | <p>The cohortDefinitionSet argument must be a data frame with the following columns:</p> <p><b>cohortId</b> The unique integer identifier of the cohort</p> <p><b>cohortName</b> The cohort's name</p> <p><b>sql</b> The OHDSI-SQL used to generate the cohort</p> <p>Optionally, this data frame may contain:</p> <p><b>json</b> The Circe JSON representation of the cohort</p> |
| stopOnError          | If an error happens while generating one of the cohorts in the cohortDefinitionSet, should we stop processing the other cohorts? The default is TRUE; when set to FALSE, failures will be identified in the return value from this function.  |
| incremental          | Create only cohorts that haven't been created before?   |

`incrementalFolder`

If `incremental = TRUE`, specify a folder where records are kept of which definition has been executed.

### Value

A data.frame consisting of the following columns:

**cohortId** The unique integer identifier of the cohort

**cohortName** The cohort's name

**generationStatus** The status of the generation task which may be one of the following:

**COMPLETE** The generation completed successfully

**FAILED** The generation failed (see logs for details)

**SKIPPED** If using `incremental == 'TRUE'`, this status indicates that the cohort's generation was skipped since it was previously completed.

**startTime** The start time of the cohort generation. If the `generationStatus == 'SKIPPED'`, the `startTime` will be NA.

**endTime** The end time of the cohort generation. If the `generationStatus == 'FAILED'`, the `endTime` will be the time of the failure. If the `generationStatus == 'SKIPPED'`, `endTime` will be NA.

---

<code>getCohortCounts</code>	<i>Count the cohort(s)</i>
------------------------------	----------------------------

---

### Description

Computes the subject and entry count per cohort

### Usage

```
getCohortCounts(
  connectionDetails = NULL,
  connection = NULL,
  cohortDatabaseSchema,
  cohortTable = "cohort",
  cohortIds = c()
)
```

### Arguments

`connectionDetails`

An object of type `connectionDetails` as created using the [createConnectionDetails](#) function in the `DatabaseConnector` package. Can be left NULL if connection is provided.

`connection`

An object of type `connection` as created using the [connect](#) function in the `DatabaseConnector` package. Can be left NULL if `connectionDetails` is provided, in which case a new connection will be opened at the start of the function, and closed when the function finishes.

`cohortDatabaseSchema`

Schema name where your cohort table resides. Note that for SQL Server, this should include both the database and schema name, for example `'scratch.dbo'`.



cohortTable	The name of the cohort table.
cohortIds	The cohort Id(s) used to reference the cohort in the cohort table. If left empty, all cohorts in the table will be included.

**Value**

A data frame with cohort counts

---

getCohortTableNames	<i>Used to get a list of cohort table names to use when creating the cohort tables</i>
---------------------	--

---

**Description**

This function creates a list of table names used by [createCohortTables](#) to specify the table names to create. Use this function to specify the names of the main cohort table and cohort statistics tables.

**Usage**

```
getCohortTableNames(
  cohortTable = "cohort",
  cohortInclusionTable = paste0(cohortTable, "_inclusion"),
  cohortInclusionResultTable = paste0(cohortTable, "_inclusion_result"),
  cohortInclusionStatsTable = paste0(cohortTable, "_inclusion_stats"),
  cohortSummaryStatsTable = paste0(cohortTable, "_summary_stats"),
  cohortCensorStatsTable = paste0(cohortTable, "_censor_stats")
)
```

**Arguments**

cohortTable	Name of the cohort table.
cohortInclusionTable	Name of the inclusion table, one of the tables for storing inclusion rule statistics.
cohortInclusionResultTable	Name of the inclusion result table, one of the tables for storing inclusion rule statistics.
cohortInclusionStatsTable	Name of the inclusion stats table, one of the tables for storing inclusion rule statistics.
cohortSummaryStatsTable	Name of the summary stats table, one of the tables for storing inclusion rule statistics.
cohortCensorStatsTable	Name of the censor stats table, one of the tables for storing inclusion rule statistics.

**Value**

A list of the table names as specified in the parameters to this function.

---

getRequiredTasks	<i>Get a list of tasks required when running in incremental mode</i>
------------------	--

---

### Description

This function will attempt to check the recordKeepingFile to determine if a list of operations have completed by comparing the keys passed into the function with the checksum supplied

### Usage

```
getRequiredTasks(..., checksum, recordKeepingFile)
```

### Arguments

...	Parameter values used to identify the key in the incremental record keeping file
checksum	The checksum representing the operation to check
recordKeepingFile	A file path to a CSV file containing the record keeping information.

### Value

Returns a list of outstanding tasks based on inspecting the full contents of the record keeping file

---

insertInclusionRuleNames	<i>Used to insert the inclusion rule names from a cohort definition set when generating cohorts that include cohort statistics</i>
--------------------------	--

---

### Description

This function will take a cohortDefinitionSet that includes the Circe JSON representation of each cohort, parse the InclusionRule property to obtain the inclusion rule name and sequence number and insert the values into the cohortInclusionTable. This function is only required when generating cohorts that include cohort statistics.

### Usage

```
insertInclusionRuleNames(
  connectionDetails = NULL,
  connection = NULL,
  cohortDefinitionSet,
  cohortDatabaseSchema,
  cohortInclusionTable = getCohortTableNames()$cohortInclusionTable
)
```

**Arguments**

connectionDetails	An object of type connectionDetails as created using the <a href="#">createConnectionDetails</a> function in the DatabaseConnector package. Can be left NULL if connection is provided.
connection	An object of type connection as created using the <a href="#">connect</a> function in the DatabaseConnector package. Can be left NULL if connectionDetails is provided, in which case a new connection will be opened at the start of the function, and closed when the function finishes.
cohortDefinitionSet	The cohortDefinitionSet argument must be a data frame with the following columns: <b>cohortId</b> The unique integer identifier of the cohort <b>cohortName</b> The cohort's name <b>sql</b> The OHDSI-SQL used to generate the cohort Optionally, this data frame may contain: <b>json</b> The Circe JSON representation of the cohort
cohortDatabaseSchema	Schema name where your cohort tables reside. Note that for SQL Server, this should include both the database and schema name, for example 'scratch.dbo'.
cohortInclusionTable	Name of the inclusion table, one of the tables for storing inclusion rule statistics.

**Value**

A data frame containing the inclusion rules by cohort and sequence ID

---

isTaskRequired	<i>Is a task required when running in incremental mode</i>
----------------	--

---

**Description**

This function will attempt to check the recordKeepingFile to determine if an individual operation has completed by comparing the keys passed into the function with the checksum supplied

**Usage**

```
isTaskRequired(..., checksum, recordKeepingFile, verbose = TRUE)
```

**Arguments**

...	Parameter values used to identify the key in the incremental record keeping file
checksum	The checksum representing the operation to check
recordKeepingFile	A file path to a CSV file containing the record keeping information.
verbose	When TRUE, this function will output if a particular operation has completed based on inspecting the recordKeepingFile.

**Value**

Returns TRUE if the operation has completed according to the contents of the record keeping file.

---

recordTasksDone	<i>Record a task as complete</i>
-----------------	----------------------------------

---

**Description**

This function will record a task as completed in the recordKeepingFile

**Usage**

```
recordTasksDone(..., checksum, recordKeepingFile, incremental = TRUE)
```

**Arguments**

...	Parameter values used to identify the key in the incremental record keeping file
checksum	The checksum representing the operation to check
recordKeepingFile	A file path to a CSV file containing the record keeping information.
incremental	When TRUE, this function will record tasks otherwise it will return without attempting to perform any action

---

saveIncremental	<i>Used in incremental mode to save values to a file</i>
-----------------	--

---

**Description**

When running in incremental mode, we may need to update results in a CSV file. This function will replace the data in fileName based on the key parameters

**Usage**

```
saveIncremental(data, fileName, ...)
```

**Arguments**

data	The data to record in the file
fileName	A CSV holding results in the same structure as the data parameter
...	Parameter values used to identify the key in the results file

# Index

computeChecksum, [2](#)  
connect, [3–8](#), [11](#)  
createCohortTables, [2](#), [9](#)  
createConnectionDetails, [3–8](#), [11](#)  
createEmptyCohortDefinitionSet, [3](#)  
  
dropCohortStatsTables, [4](#)  
  
exportCohortStatsTables, [4](#)  
  
generateCohort, [5](#)  
generateCohortSet, [6](#)  
getCohortCounts, [8](#)  
getCohortTableNames, [3–5](#), [9](#)  
getRequiredTasks, [10](#)  
  
insertInclusionRuleNames, [10](#)  
isTaskRequired, [11](#)  
  
recordTasksDone, [12](#)  
  
saveIncremental, [12](#)