Runing Cohort Diagnostics in an OHDSI Package mode

Gowtham A. Rao

2021-11-13

Contents

1 Introduction

There are currently two approaches to run Cohort Diagnostics. - Embed in an OHDSI study package, where all the cohort definitions are stored as part of that study package, or - WebAPi mode - where cohort diagnostics dynamically pulls the cohort definition from a webapi instance. WebAPI is the backend of the OHDSI ATLAS application, allowing programmatic access to the cohort definitions created in ATLAS.

This vignette describes how to create a self contained shareable study package that can execute cohort diagnostics on a given set of cohort definitions. This is the recommended mode to run CohortDiagnostics for a study.

In this vignette, you will learn how to - use the Cohort Diagnostics template package - Hydrate the template package with the cohort definitions you want to diagnose. - Execute the diagnostics on one or more data sources. - Review the result set using the Diagnostics Explorer R shiny app of Cohort Diagnostics - (optional) Submit the results of your cohort diagnostics to the OHDSI Phenotype library.

1.1 Assumptions:

- You have a set of valid cohort definition sql files (in parameterized OHDSI Sql format) that you want to run diagnostics on, or you are able to export cohort definitions that you want to diagnose from Atlas by using the export tab within cohort definition module of Atlas or using ROhdsiWebApi as described here.
- You have access to person level data in OMOP CDM V5.x + format on a database, are able to execute and instantiate cohorts in cohort tables (i.e. you have access to a schema in the same database with Create, Read, Update, Delete privileges.)

2 Creating the study package

The cohorts to diagnose should have two attributes to them.

- **id**: A unique integer identifier for the cohort you are diagnosing. If you created your cohort using atlas, this is your atlas id.
- **cohortName**: The unique string name for your cohort. Usually it is the same as your Atlas cohort name, but it can be any name you wish to identify your cohort. This name will be used to display your cohort in the Diagnostics Explorer shiny app.

Example:

id	cohortName
17561	[PL 4112853001] Malignant tumor of breast referent concept incident cohort: First occurrence of referent concept + descendants with >=365d prior observation

2.1 option A: Using Hydra and ROhdsiWebApi

Note: this option is required Hydra 0.3, as of writing this document that version of Hydra is in develop branch.

The skeleton cohort diagnostics study package is here. A skeleton package is a special package that is designed to be used with Hydra. The input for Hydra is study a specifications file (in json format). Example is here.

```
# please ensure you have the latest version of Hydra. As of 08/13/2021 - CohortDiagnostics support for
# please check hydra release notes and update hydra
remotes::install_github("OHDSI/Hydra", ref = "develop")
outputFolder <- "d:/temp/output"  # location where you study package will be created


########## Please populate the information below #####################
version <- "v0.1.0"
name <- "Thrombosis With Thrombocytopenia Syndrome cohorts - an OHDSI network study"
packageName <- "ThrombosisWithThrombocytopeniaSyndrome"
skeletonVersion <- "v0.0.1"
createdBy <- "rao@ohdsi.org"
createdDate <- Sys.Date() # default
modifiedBy <- "rao@ohdsi.org"
modifiedDate <- Sys.Date()
skeletonType <- "CohortDiagnosticsStudy"
organizationName <- "OHDSI"
description <- "Cohort diagnostics on Thrombosis With Thrombocytopenia Syndrome cohorts."


library(magrittr)
# Set up
baseUrl <- Sys.getenv("baseUrl")
# if you have security enabled, please authorize the use - example below
# ROhdsiWebApi::authorizeWebApi(baseUrl, 'windows')
cohortIds <- c(22040,
               22042,
               22041,
               22039,
               22038,
               22037,
               22036,
               22035,
               22034,
               22033,
               22031,
               22032,
               22030,
               22028,
```

```r
               22029)



################# end of user input ##############
webApiCohorts <- ROhdsiWebApi::getCohortDefinitionsMetaData(baseUrl = baseUrl)
studyCohorts <-  webApiCohorts %>%
        dplyr::filter(.data$id %in% cohortIds)

# compile them into a data table
cohortDefinitionsArray <- list()
for (i in (1:nrow(studyCohorts))) {
        cohortDefinition <-
                ROhdsiWebApi::getCohortDefinition(cohortId = studyCohorts$id[[i]],
                                                  baseUrl = baseUrl)
        cohortDefinitionsArray[[i]] <- list(
                id = studyCohorts$id[[i]],
                createdDate = studyCohorts$createdDate[[i]],
                modifiedDate = studyCohorts$createdDate[[i]],
                logicDescription = studyCohorts$description[[i]],
                name = stringr::str_trim(stringr::str_squish(cohortDefinition$name)),
                expression = cohortDefinition$expression
        )
}

tempFolder <- tempdir()
unlink(x = tempFolder, recursive = TRUE, force = TRUE)
dir.create(path = tempFolder, showWarnings = FALSE, recursive = TRUE)

specifications <- list(id = 1,
                       version = version,
                       name = name,
                       packageName = packageName,
                       skeletonVersion = skeletonVersion,
                       createdBy = createdBy,
                       createdDate = createdDate,
                       modifiedBy = modifiedBy,
                       modifiedDate = modifiedDate,
                       skeletonType = skeletonType,
                       organizationName = organizationName,
                       description = description,
                       cohortDefinitions = cohortDefinitionsArray)

jsonFileName <- paste0(file.path(tempFolder, "CohortDiagnosticsSpecs.json"))
write(x = specifications %>% RJSONIO::toJSON(pretty = TRUE, digits = 23), file = jsonFileName)


##############################################################
##############################################################
#######       Get skeleton from github            ############
#######       Uncomment if you want to use latest ############
#######       skeleton only - for advanced user   ############
##############################################################
```

```
##############################################################
##############################################################
#### get the skeleton from github
# download.file(url = "https://github.com/OHDSI/SkeletonCohortDiagnosticsStudy/archive/refs/heads/main.
#                          destfile = file.path(tempFolder, 'skeleton.zip'))
# unzip(zipfile =  file.path(tempFolder, 'skeleton.zip'),
#       overwrite = TRUE,
#       exdir = file.path(tempFolder, "skeleton")
#         )
# fileList <- list.files(path = file.path(tempFolder, "skeleton"), full.names = TRUE, recursive = TRUE,
# DatabaseConnector::createZipFile(zipFile = file.path(tempFolder, 'skeleton.zip'),
#                                  files = fileList,
#                                  rootFolder = list.dirs(file.path(tempFolder, 'skeleton'), recursive

##############################################################
##############################################################
#######              Build package             #############
##############################################################
##############################################################
##############################################################


#### Code that uses the ExampleCohortDiagnosticsSpecs in Hydra to build package
hydraSpecificationFromFile <- Hydra::loadSpecifications(fileName = jsonFileName)
unlink(x = outputFolder, recursive = TRUE)
dir.create(path = outputFolder, showWarnings = FALSE, recursive = TRUE)
Hydra::hydrate(specifications = hydraSpecificationFromFile,
               outputFolder = outputFolder
)

# for advanced user using skeletons outside of Hydra
# Hydra::hydrate(specifications = hydraSpecificationFromFile,
#                outputFolder = outputFolder,
#                skeletonFileName = file.path(tempFolder, 'skeleton.zip')
# )


unlink(x = tempFolder, recursive = TRUE, force = TRUE)
```