R package: Cohort2Trajectory

Markus Haug

09.09.2022

Guide for running Cohort2Trajectory

The code for running the package is located in "./extras/CodeToRun.R".

1. Install & load the package

```
devtools::install_github("HealthInformaticsUT/Cohort2Trajectory")
library(Cohort2Trajectory)
```

2. Set up the study settings & database credentials

```
studyName <- "HeartFailure"
```

The variable *studyName* can be whatever and has two purposes:
1. Saving all the results and configurations with the corresponding prefix.
2. Loading study settings if such study is saved in "./inst/Settings/trajectorySettings.csv".

The connecting user must have:
1. Select permissions on Common Data Model data and results schemas.
2. Select and create permissions on Common Data Model temp schema.

```
pathToResults <- getwd()
pathToDriver <- './Drivers'
dbms <- "postgresql"
user <- 'user'
pw <- "password"
server <- 'localhost/test_database'
port <- '5432'

cdmSchema <- "ohdsi_cdm"
cdmTmpSchema <- "ohdsi_temp"
cdmResultsSchema <- "ohdsi_results"
baseUrl <- "http://localhost:8080/WebAPI"
```

These variables are used to create a connection with a local or remote database server. The variables *pathToResults* and *pathToDriver* can be customized as preferred, but keep in mind that the drivers path

has to include your database management system (DBMS) driver. These set variable values would create a connection with PostgreSQL DMBS database called 'test_database' located on localhost running on port 5432. The user connecting is 'user' with password 'password'. The relevant OHDSI CDM schemas are 'ohdsi_cdm', 'ohdsi_results' and for temporary tables – 'ohdsi_temp'. The WebAPI runs on 'http://localhost:8080/WebAPI' (this is only needed if the study has not yet been defined).

3. Create connection with the defined database

```
connectionDetails <-
  DatabaseConnector::createConnectionDetails(
    dbms = dbms,
    server = server,
    user = user,
    password = pw,
    port = port,
    pathToDriver = pathToDriver
  )

conn <- DatabaseConnector::connect(connectionDetails)
```

4. Running the package

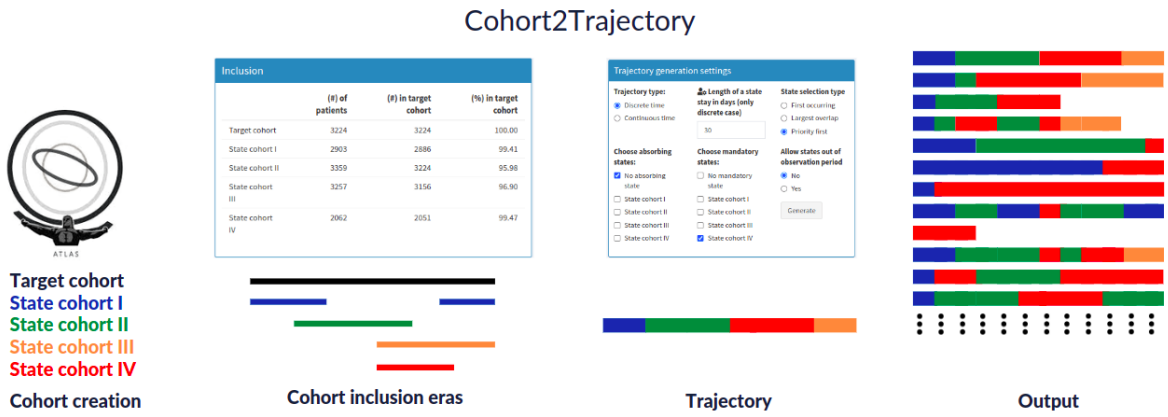

Figure 1: Summary of Cohort2Trajectory

Keep in mind that before running the package the relevant cohorts in interest have to be generated and saved in results schema (not neccessary when a study has been defined and JSONs or SQLs saved in "./inst/"). The package can be run with four different function calls.

4.1 Running the package in GUI

Running the package in a GUI is advised when unfamiliar with the package as it has some guidelines integrated.

There are 8 tabs in the GUI:
1. Description
2. Import via Atlas - Import cohorts with cohort ids defined in ATLAS.

3. Import via JSON - Import cohorts with JSONs saved in "./inst/JSON/".
4. Statistics - Some preliminary statistics about the imported cohorts and generated trajectories.
5. Prioritization - A tab for prioritizing states for resolving merge conflicts (first one having the highest priority).
6. Trajectories - A tab for configuring study settings.
7. Profiles - Check the trajectories of individual patients in the generated trajectories.
8. Help - A tab with some useful tips.

```
runGUI(
  conn,
  connectionDetails,
  pathToDriver = pathToDriver,
  pathToResults = pathToResults,
  dbms = dbms,
  cdmSchema = cdmSchema,
  cdmTmpSchema = cdmTmpSchema,
  cdmResultsSchema = cdmResultsSchema,
  studyName = studyName,
  baseUrl = baseUrl
)
```

The GUI is interactive.

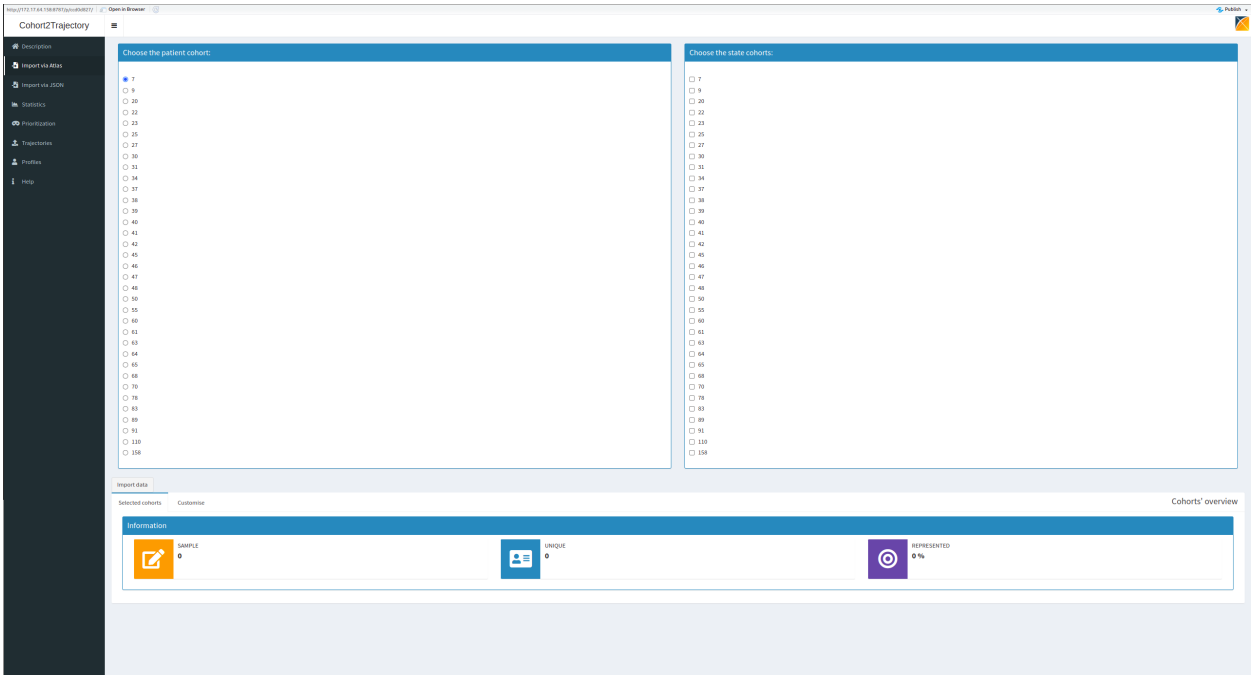

Figure 2: Screenshot of a cohort import tab

4.2 Running the package in CLI

Everything that can be done in the GUI can also be done in CLI. The values of the relevant variables have to be defined.

```
Cohort2Trajectory(
  dbms = dbms,
  connection = conn,
  cdmSchema = cdmSchema,
  cdmTmpSchema = cdmTmpSchema,
  cdmResultsSchema = cdmResultsSchema,
  studyName = studyName,
  baseUrl = baseUrl,
  atlasTargetCohort = 1, # Target cohort id from ATLAS.
  atlasStateCohorts = c(2,3,4), # State cohorts' ids from ATLAS.
  # Customized labels in import order.
  stateCohortLabels = c("State2", "State3", "State4"),
  # Priority order of states.
  stateCohortPriorityOrder = c("State4", "State3", "State2"),
  stateCohortMandatory = c("State2"), # Mandatory states
  stateCohortAbsorbing = c("State4"), # Absorbing states
  ##############################################################################
  # stateSelectionTypes
  # 1 - First occurring
  # 2 - Largest overlap
  # 3 - Priority ordering
  ##############################################################################
  stateSelectionType = 3,
  ##############################################################################
  # trajectoryType
  # 0 - Discrete time
  # 1 - Continuous time
  ##############################################################################
  trajectoryType = 0,
  lengthOfStay = 30, # 30 days
  outOfCohortAllowed = TRUE, # We can allow (TRUE) the
  # imported state cohorts to be included if they occur
  # after the observation period of the target cohort.
  runSavedStudy = FALSE, # If we run a saved study
  # the study configuration is not needed to
  # set explicitly (see below).
  pathToResults = pathToResults
)
```

4.3 Running the package using predefined study in CLI

As mentioned before, study settings are saved in "./inst/Settings/trajectorySettings.csv". For alternating between defined studies the corresponding *studyName* variable has to be defined.

```
Cohort2Trajectory(
  dbms = dbms,
  connection = conn,
  cdmSchema = cdmSchema,
  cdmTmpSchema = cdmTmpSchema,
  cdmResultsSchema = cdmResultsSchema,
  studyName = studyName,
```

```
  runSavedStudy = TRUE,
  pathToResults = pathToResults
)
```

4.4 Running the package without connecting to the database

There is also a possibility to run the package without connecting to any OMOP CDM databases. This assumes that the user already has a .csv file which consists of target cohorts and state cohorts. An easy example of the .csv file is given below.

```
"SUBJECT_ID","COHORT_DEFINITION_ID","COHORT_START_DATE","COHORT_END_DATE"
3141,"0",2020-02-22,2021-02-23
3150,"State1",2020-11-30,2020-12-01
3150,"State2",2021-01-01,2021-02-01
...,...,...,...
```

The columns are as follows:
1. SUBJECT_ID (unique for each patient)
2. COHORT_DEFINITION_ID (the target cohort has to be denoted as "0". Other, state cohorts, can be denoted in any way convenient (names should make sense))
3. COHORT_START_DATE (The start date of the corresponding cohort)
4. COHORT_END_DATE (The end date of the corresponding cohort)

Code for running the package:

```
Cohort2Trajectory(
  studyName = studyName,
  stateCohortPriorityOrder = c("State1", "State3", "State2"), # Priority order of states
  stateCohortMandatory = c("State2"), # Mandatory states
  stateCohortAbsorbing = c("State3"), # Absorbing states
  ################################################################################
  # stateSelectionTypes
  # 1 - First occurring
  # 2 - Largest overlap
  # 3 - Priority ordering
  ################################################################################
  stateSelectionType = 3,
  ################################################################################
  # trajectoryType
  # 0 - Discrete time
  # 1 - Continuous time
  ################################################################################
  trajectoryType = 1,
  lengthOfStay = 30, # Only relevant when using "discrete" case
  outOfCohortAllowed = TRUE,  # We can allow (TRUE) the
  # imported state cohorts to be included if they occur
  # after the observation period of the target cohort.
  runSavedStudy = FALSE,
  pathToResults = pathToResults,
  useCDM = FALSE, # Has to be false when running without a connection to OMOP CDM
```

```
  pathToData = paste(getwd(),'/tmp/datasets/importedData.csv', sep = "") # Path to the data file.
)
```

Outputs

The workflow creates a .csv file with the prefix being the *studyName* variable and the suffix depending on the trajectory creation configuration. The output will reside in the path of variable *pathToResults* inside the directory *./tmp/datasets/*. A very simple output example:

```
"SUBJECT_ID","STATE","STATE_START_DATE","STATE_END_DATE","TIME_IN_COHORT","GEND..."
1,"START",1982-08-18,1982-08-18,0,8507,33.539,"1"
1,"State1",1982-08-19,1982-08-19,0.022,8507,33.539,"2"
1,"State1",1982-08-20,1982-08-20,0.044,8507,33.561,"2"
1,"State2",1982-08-21,1982-08-21,0.066,8507,33.583,"3"
1,"State2",1982-08-22,1982-08-22,0.088,8507,33.605,"3"
1,"State3",1982-08-23,1982-08-23,0.110,8507,33.627,"4"
1,"State2",1982-08-24,1982-08-24,0.132,8507,33.649,"3"
1,"State1",1982-08-25,1982-08-25,0.154,8507,33.671,"2"
1,"EXIT",1982-08-12,1982-08-26,0.176,8507,33.693,"5"
2,"START",2009-05-30,2009-05-30,0,8507,40.849,"1"
2,"State1",2009-05-31,2009-05-31,0.22,8507,40.871,"2"
2,"State1",2009-06-01,2009-06-01,0.44,8507,40.893,"2"
2,"State2",2005-06-02,2005-06-02,0.66,8532,40.915,"3"
2,"State1",2005-06-03,2005-06-03,0.88,8532,40.937,"2"
2,"State1",2005-06-04,2005-06-04,0.110,8532,40.959,"2"
2,"EXIT",2009-06-05,2009-06-05,0.132,8507,40.981,"5"
3,"START",2011-01-30,2011-01-30,0,8507,45.021,"1"
...,...,...,...,...,...,...,...
```

The columns of the .csv file are “SUBJECT_ID”,“STATE”,“STATE_START_DATE”, “STATE_END_DATE”,“TIME_IN_COHORT”,“GENDER_CONCEPT_ID”,“AGE”,“STATE_ID”.

For more information see package manual, source code or contact the maintainer.