Trajectories

Kadri Künnapuu Kadri Ligi Raivo Kolde Sven Laur Solomon Ioannou
Peter Rijnbeek Jaak Vilo Sulev Reisberg

Introduction

This vignette describes how to use the Trajectories package in discovery and validation mode. The discovery mode enables to discover and visualize event pairs in OMOP-formatted observational health data while the validation mode validates the results against event pairs discovered elsewhere.

We have selected type 2 diabetes as an example, creating graphs with both conditions and drug eras included in the analysis.

The package can be run in 2 modes: discovery and validation. Both modes have separate CodeToRun…R files in the `extras` folder.

Installation instructions

1. Clone the source code from https://github.com/EHDEN/Trajectories repository
2. Open the project (file ***Trajectories.Rproj***) in RStudio
3. Install the package via top-right menu: ***Build*** -> ***Install and Restart***

Running the package

In order to run the package, you need:

1. A database that has data in OMOP CDM v5 format. The database should also contain OMOP vocabulary, but this can be in a separate schema.
2. A database user + passwords that has:
   - Read (SELECT) permission from OMOP CDM tables and vocabulary
   - CREATE, DROP, SELECT, INSERT, UPDATE, DELETE permission in some schema of the same database. This is used for creating and temporary analysis tables. For this, you can create a separate schema in the same database.

In case you do not have an access to any OMOP CDM, you can use test data from Eunomia package. To run a package on that data, simply run all commands from `./extras/CodeToRunEunomia.R`

Setting up the database login credentials

As the package needs to connect to the database, you have to add login credentials into ***.Renviron*** file:

1. Rename ***Renviron.template*** to ***.Renviron***

2. Add your database username and password to ***.Renviron*** (these are the database credentials for accessing OMOP CDM and writing temporary analysis tables/data). After you restart your RStudio, it automatically reads the database credentials from that file so that you do not have to enter them each time. Also, .Renviron is not under version control, therefore it is kept unchanged even if you pull the updates of the R-package.
3. Restart RStudio to automatically read in database credentials.

Defining the study cohort and analysis settings

The package searches for event trajectories within cohort. This means that you either ***have to define a cohort by yourself*** or ***use some built-in cohort*** (or a cohort defined by someone else).

The built-in cohorts + analysis settings are located in `./inst/extdata/` folder. For example, the validation of the event pairs of Type 2 diabetes cohort can be found from `./inst/extdata/T2D-validate`. If you are going to use built-in setup, skip the rest of this section

If you need to create a new cohort + analysis, follow these steps:

1. Create an empty folder
2. Define the cohort in your Atlas server (https://atlas.ohdsi.org/)
3. Export SQL of the cohort definition (format "SQL Server") to `cohort.sql` file and save it to your created folder.
4. Take `./inst/extdata/T2D/trajectoryAnalysisArgs.json` file and copy it to the same folder.
5. Modify the parameters (analysis settings) in `trajectoryAnalysisArgs.json` to fit your needs.

For more information of the parameters, run

`?Trajectories::createTrajectoryAnalysisArgs`

Setting up study parameters and running the package

The easiest way to run the package is to open either `./extras/CodeToRunDiscover.R` or `./extras/CodeToRunValidate.R` depending on whether you wish to find new event trajectories or you are just validating the pairs from someone's result.

In order to validate the event pairs of Type 2 Diabetes that were reported by Kunnapuu et al., open `./extras/CodeToRunValidate.R`.

You have to edit some lines in the file before running it. As both files are quite similar, so the guidelines are common.

First, add correct connection string to

```
connectionString = "jdbc:postgresql://[host]:[port]/[database]"
```

Second, set up the correct values when calling `createTrajectoryLocalArgs()`

```
oracleTempSchema = "temp_schema", #no need to change this even when you are not using Oracle
prefixForResultTableNames = "", #It is a prefix that is used for creating temporary table names.
              #You can be set to "".
cdmDatabaseSchema = '...', #It is the name of database schema where the OMOP CDM data is actually
              #kept/taken
vocabDatabaseSchema = '...', #It is the name of database schema where the OMOP vocabularies are
              #kept/taken
```

```
resultsSchema = '...', #It is the name of database schema where the analysis tables will be
             #created
sqlRole = F, #You may always use 'F'. Setting specific role might be useful in PostgreSQL when
             #you want to create tables by using specific role so that the others also see the
             #results. However, then you must ensure that this role has permissions to read from
             #all necessary schemas and also create tables to resultsSchema
inputFolder=system.file("extdata", "T2D-validate", package = "Trajectories"), # Full path to
             #input folder that contains SQL file for cohort definition and
             #trajectoryAnalysisArgs.json. You can use built-in folders of this package such as:
             #inputFolder=system.file("extdata", "T2D-validate", package = "Trajectories").
             #Otherwise use full path: inputFolder='/here/is/my/path'
mainOutputFolder='/here/is/my/path', #Path to general folder where all the outputs of the package
             #will be written. The folder must exist. Each database and analysis will have a
             #separate subfolder automatically.
databaseHumanReadableName='...' #Use something short. This will be used as a folder name an it will
             #be added to the titles of the graph.
```

For more information about the parameters you can always use

```
?createTrajectoryLocalArgs
```

In order to validate the event pairs of Type 2 Diabetes that were reported by Kunnapuu et al., use

```
inputFolder=system.file("extdata", "T2D-validate", package = "Trajectories")
```

Now you have all set and you can simply run the workhorse `Trajectories::discover()` or `Trajectories::validate()`.

Both processes are made of several subprocesses which start one after another. For your convenience, you can change TRUE/FALSE flags when calling these subprocesses. Run

```
?Trajectories::discover
```

or

```
?Trajectories::validate
```

for more information!

Assessing the results

All results are created to the folder that was specified by parameters ***mainOutputFolder*** and ***databaseHumanReadableName***.

Note that subfolder ***validation_setup*** is always created. This is the folder that you can pass to someone else to validate your results in their database - it contains both the cohort definition and analysis settings.