R package: TrajectoryMarkovAnalysis

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

```r
devtools::install_github("HealthInformaticsUT/TrajectoryMarkovAnalysis")
library(TrajectoryMarkovAnalysis)
```

2. Set up the study settings & database credentials

```r
studyName <- "HeartFailure"
databaseDescription <- "This is a very cool database."
```

The variable *studyName* can be whatever and has two purposes:
1. Saving all the results with the corresponding prefix.
2. Comparing different results in a handy GUI.

The variable *databaseDescription* should describe the database the study is run on. This will be displayed in the dashboard for comparing different databases.

```r
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"
```

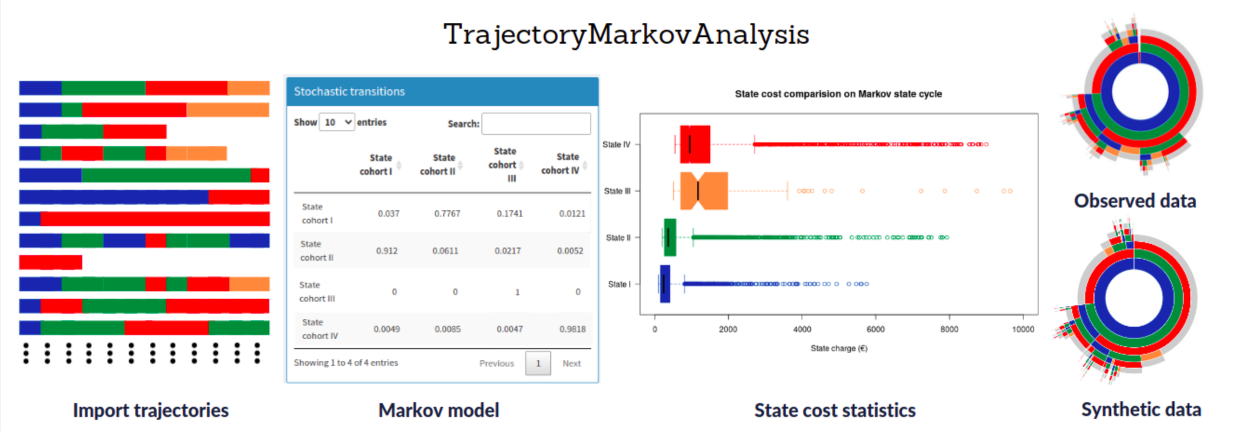
These variables are used to create a connection with a local or remote database server. The variables *pathToResults* and *pathToDriver* can be customized as preferred, but keep in mind that the drivers path has to include your database management system (DBMS) driver. These set variable values would 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'.

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



	State cohort I	State cohort II	State cohort III	State cohort IV
State cohort I	0.037	0.7767	0.1741	0.0121
State cohort II	0.912	0.0611	0.0217	0.0052
State cohort III	0	0	1	0
State cohort IV	0.0049	0.0085	0.0047	0.9818

Figure 1: Summary of TrajectoryMarkovAnalysis

Keep in mind that before running the package you need access to a .csv file which has been created with *Cohort2Trajectory* package or a .csv file with same features. The package can be used with five different functions.

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 - Import the patient trajectories .csv file.
3. Discrete time Markov analysis - Create a discrete time Markov chain and querying state cost data.
4. Continuous time Markov analysis - Create a continuous time Markov model and querying state cost data.

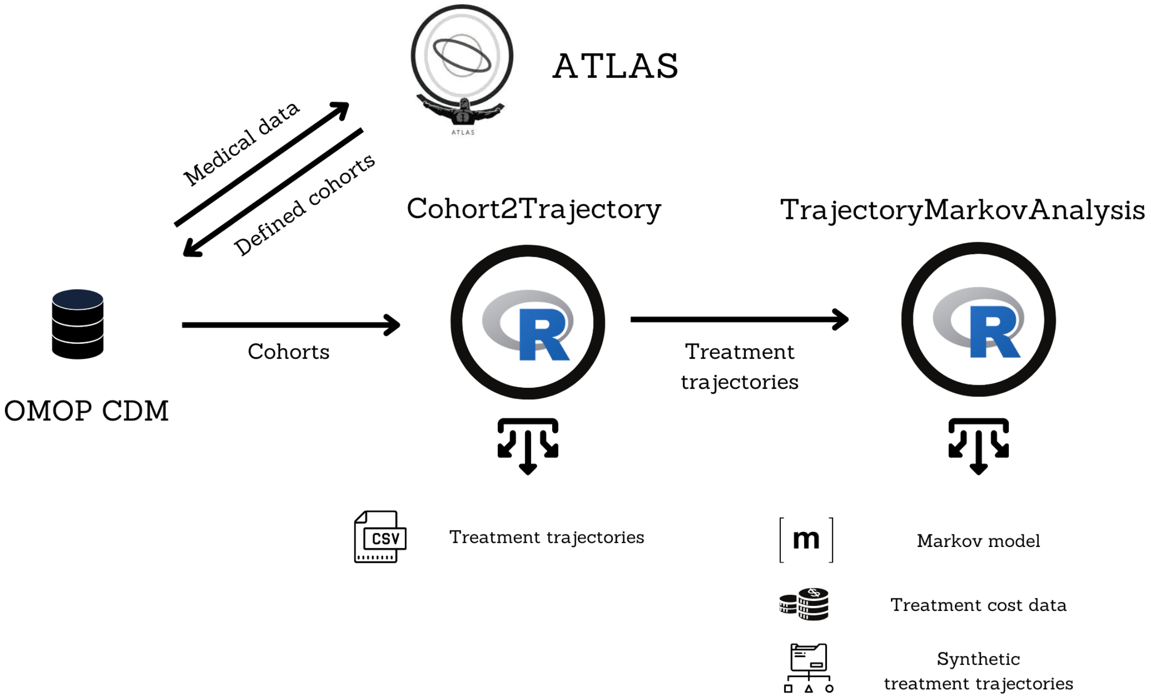
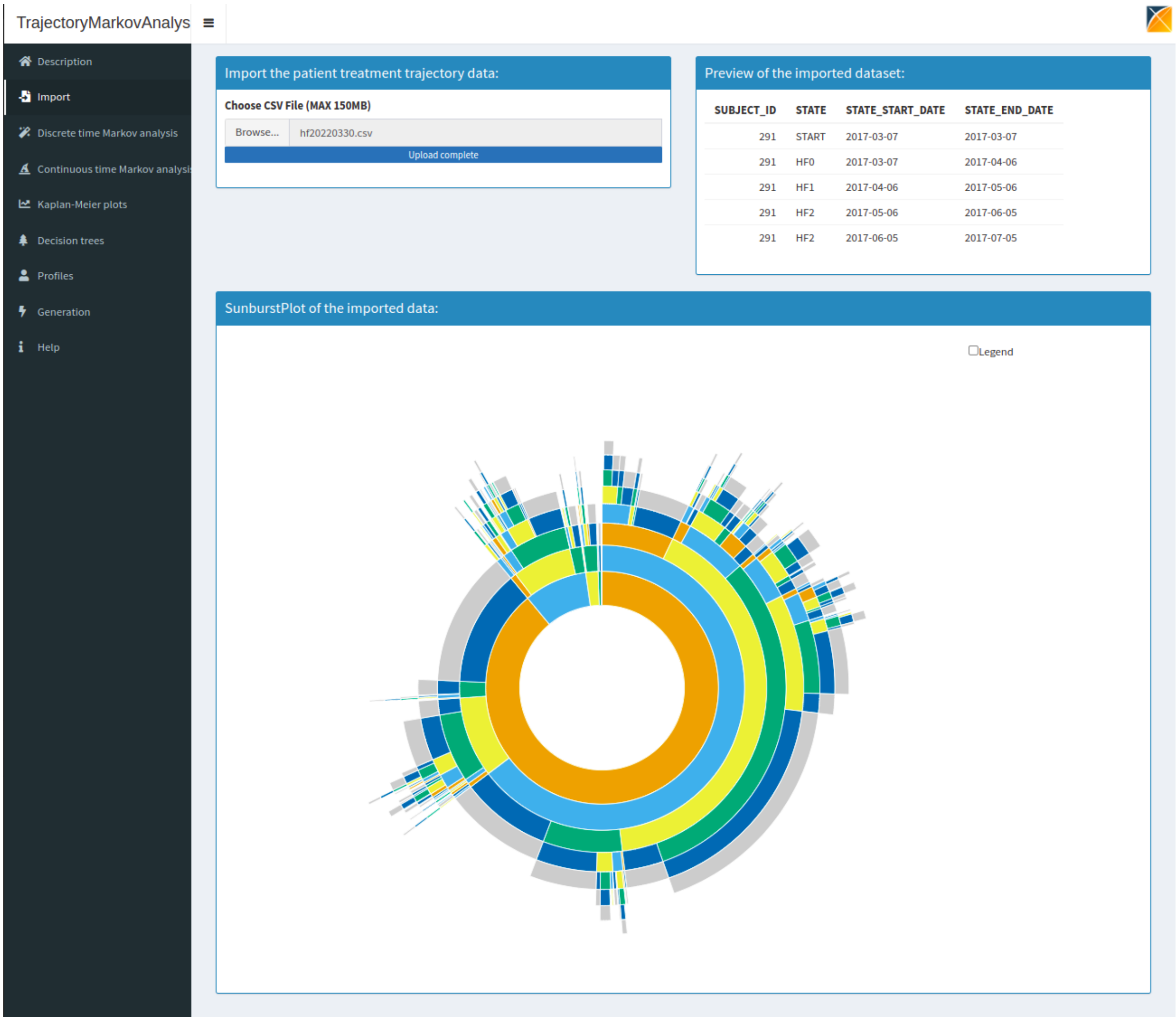


Figure 2: Suggested workflow: using Cohort2Trajectory and TrajectoryMarkovAnalysis

5. Kaplan-Meier plots - A tab for creating Kaplan-Meier plots.
6. Decision trees - A tab for creating trajectory trees.
7. Profiles - Check the trajectories of individual patients in the imported trajectories.
8. Generation - A tab with some useful tips.
9. Help - A tab with some useful tips.

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

The GUI is interactive.



Import the patient treatment trajectory data:

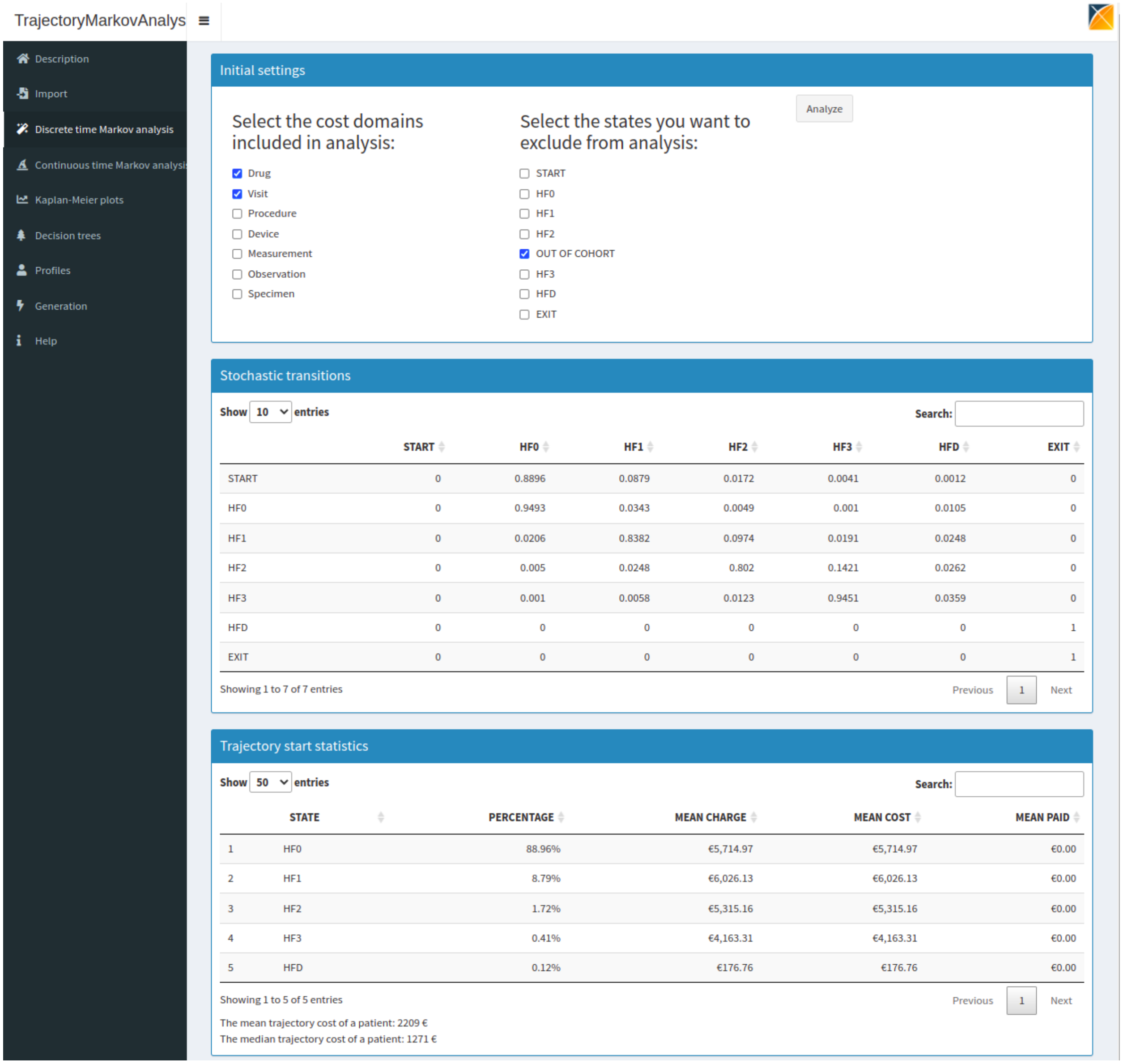
Choose CSV File (MAX 150MB)

hf20220330.csv

Preview of the imported dataset:

SUBJECT_ID	STATE	STATE_START_DATE	STATE_END_DATE
291	START	2017-03-07	2017-03-07
291	HF0	2017-03-07	2017-04-06
291	HF1	2017-04-06	2017-05-06
291	HF2	2017-05-06	2017-06-05
291	HF2	2017-06-05	2017-07-05

SunburstPlot of the imported data:



Initial settings

Select the cost domains included in analysis:

☑ Drug
☑ Visit
□ Procedure
□ Device
□ Measurement
□ Observation
□ Specimen

Select the states you want to exclude from analysis:

□ START
□ HF0
□ HF1
□ HF2
☑ OUT OF COHORT
□ HF3
□ HFD
□ EXIT

Stochastic transitions

Show 10 entries Search:

	START	HF0	HF1	HF2	HF3	HFD	EXIT
START	0	0.8896	0.0879	0.0172	0.0041	0.0012	0
HF0	0	0.9493	0.0343	0.0049	0.001	0.0105	0
HF1	0	0.0206	0.8382	0.0974	0.0191	0.0248	0
HF2	0	0.005	0.0248	0.802	0.1421	0.0262	0
HF3	0	0.001	0.0058	0.0123	0.9451	0.0359	0
HFD	0	0	0	0	0	0	1
EXIT	0	0	0	0	0	0	1

Showing 1 to 7 of 7 entries Previous 1 Next

Trajectory start statistics

Show 50 entries Search:

	STATE	PERCENTAGE	MEAN CHARGE	MEAN COST	MEAN PAID
1	HF0	88.96%	€5,714.97	€5,714.97	€0.00
2	HF1	8.79%	€6,026.13	€6,026.13	€0.00
3	HF2	1.72%	€5,315.16	€5,315.16	€0.00
4	HF3	0.41%	€4,163.31	€4,163.31	€0.00
5	HFD	0.12%	€176.76	€176.76	€0.00

Showing 1 to 5 of 5 entries Previous 1 Next

The mean trajectory cost of a patient: 2209 €
The median trajectory cost of a patient: 1271 €

4.2 Running the package in CLI

All the results created in GUI can be also be done in CLI. The values of the relevant variables have to be defined.

```
inputData <- readr::read_csv("/home/...") # trajectory data
modelType <- "discrete" # "discrete" or "continuous"
# One can exclude any of the states present in the trajectory data
excludedStates <- c("OUT OF COHORT")

# Selection of which cost data domains will be included
costDomains <- c(
                'Drug',
                'Visit',
                'Procedure',
                'Device',
                'Measurement',
                'Observation',
                'Specimen'
                )

TrajectoryMarkovAnalysis(
 conn,
 dbms,
 cdmSchema,
 cdmTmpSchema,
 inputData,
 modelType,
 studyName,
 pathToResults,
 excludedStates,
 costDomains,
 databaseDescription
)
```

4.3 Generating discrete trajectories from CLI

After generating Markov discrete models you can create synthetic data based on the learned Markov process. A corresponding .csv file will be created.
NOTE: This can also be done using the GUI!

```
# Import the relevant model
transistionMatrix = get(load(paste(pathToResults,"/tmp/models/..." ,sep = "")))

generateDataDiscrete(transitionMatrix = transistionMatrix,
                               n = 100, # Number of patients
                               minDate = "1900-01-01",
                               maxDate = "2021-12-31",
                               maxOut = 365, # Maximum days out of cohort
                               stateDuration = 30, # state duration (time in days)
                               pathToResults = getwd(),
                               studyName = studyName)
```

4.4 Generating continuous trajectories from CLI

After generating Markov discrete models you can create synthetic data based on the learned Markov process. A corresponding .csv file will be created.
NOTE: This can also be done using the GUI!

```
# Import the relevant intensity matrix
intensityMatrix = get(load(paste(pathToResults,"/tmp/models/..." ,sep = "")))

generateDataContinuous(model = intensityMatrix,
                      n = 100,  # Number of patients
                      minDate = "1900-01-01",
                      maxDate = "2021-12-31",
                      pathToResults = pathToResults,
                      studyName = studyName)
```

Run a dashboard for comparing results from different databases or different target cohorts

NOTE: The dashboard for comparing databases is still very much in production and new features will be added soon.
The results are accessible from "./tmp/databases/". The pathToResults variable should point to the directory with subfolders "tmp/databases/…" which will be created as the result of running *TrajectoryMarkovanalysis* package.

```
runDashboard(pathToResults = pathToResults)
```

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