Introduction to CohortPathways

Gowtham A. Rao

March 24, 2025

Contents

1 Introduction

The **CohortPathways** package provides functionality to compute and visualize the sequence of events (cohorts) that occur after a person enters a specific target cohort. This is often called a *pathway* or *treatment pathway*. The concepts here closely align with the Cohort Pathways functionality in ATLAS, allowing you to replicate or extend those analyses in a scripted environment.

Key highlights:

- **Compute Pathways**: Calculate how often different sequences of events occur among patients in one or more target cohorts.
- **Parameterization**: Adjust `allowRepeats`, `collapseWindow`, `maxDepth`, etc., to customize how events are grouped or truncated.
- **Summaries**: Retrieve summary statistics and detailed breakdowns of event sequences to facilitate advanced analyses or visualizations (like sunburst plots).

In this vignette, we'll walk through a typical workflow: from installing **CohortPathways** to running an example of pathway analysis.

1.1 Installation

You can install the latest development version of **CohortPathways** from Cran:

```
install.packages("CohortPathways")
```

```r
library(CohortPathways)
```

1.2 Setting Up: Database Connection

Most functionality in **CohortPathways** requires a connection to your OMOP Common Data Model (CDM) instance and a corresponding results schema with instantiated cohorts. You'll typically do something like this:

```r
library(DatabaseConnector)

connectionDetails <- createConnectionDetails(
  dbms     = "postgresql",
  server   = "myserver/mydatabase",
  user     = "username",
  password = "password"
)
```

- If you already have an established connection (via `connect()`), you can pass that into the relevant functions directly.

1.3 Running a Simple Pathway Analysis

1.3.1 Step 1: Identify or Instantiate Cohorts

You need **target cohorts** (the index or initial condition) and **event cohorts** (the subsequent events of interest). These cohorts should already exist in a table (often named `"cohort"`) within your results schema.

For example, suppose you have:

- **Target Cohort IDs**: 101, 102 (maybe "Type 2 Diabetes" and "Hypertension")
- **Event Cohort IDs**: 201, 202, 203 (maybe different medication cohorts)

These cohorts are stored in `my_results_schema.my_cohort_table`.

1.3.2 Step 2: Execute the Pathway Analysis

Below is a minimal example using the function `executeCohortPathways()`:

```r
result <- executeCohortPathways(
  connectionDetails      = connectionDetails,
  # OR you could pass connection = myExistingConnection,
  cohortDatabaseSchema   = "my_results_schema",
  cohortTableName        = "my_cohort_table",
  targetCohortIds        = c(101, 102),
  eventCohortIds         = c(201, 202, 203),
  minCellCount           = 5,
  allowRepeats           = TRUE,
  maxDepth               = 5,
  collapseWindow         = 30
)
```

```r
names(result)
# [1] "pathwayAnalysisStatsData"   "pathwaysAnalysisPathsData"
# [3] "pathwaysAnalysisEventsData" "pathwaycomboIds"
# [5] "pathwayAnalysisCodesLong"   "isCombo"
# [7] "pathwayAnalysisCodesData"
```

- **`allowRepeats = TRUE`** means the same event cohort can appear multiple times in a pathway.
- **`collapseWindow = 30`** means events occurring within 30 days of each other are considered part of the same step.
- **`maxDepth = 5`** means each pathway is truncated at 5 steps (e.g., if a patient had 8 different events, we only keep the first 5 in the final pathway sequence).

1.3.3 Step 3: Explore the Results

The `result` object is a named list of data frames. For instance:

1. **`result$pathwayAnalysisStatsData`**
   Summaries of how many persons are in each target cohort, how many events were observed, etc.
2. **`result$pathwaysAnalysisPathsData`**
   The distinct sequences (or "paths") of events. Each row typically includes a set of columns like `step1`, `step2`, ... up to `maxDepth`, and counts of how many people followed that path.
3. **`result$pathwaysAnalysisEventsData`**
   A more granular breakdown of the events in each path.

You can use these data frames to build your own custom visualizations, or feed them into an R-based sunburst library or Sankee plot.

For example, to see the top 10 most frequent pathways:

```r
library(dplyr)

result$pathwaysAnalysisPathsData %>%
  arrange(desc(personCount)) %>%
  head(10)
```

You might see something like:

step1	step2	step3	personCount	...
201	202	NA	250	...
201	203	203	100	...
...	...	...	...	...

(Where `201` might represent a certain medication, etc.)

1.4 Practical Considerations

1. **Database Requirements**:
   You must have a results schema with the relevant cohorts instantiated. That means they must appear in the table you've specified via `cohortDatabaseSchema` and `cohortTableName`.

2. **Performance**:
   Pathway analyses can involve large intermediate tables. For big cohorts, consider using a sufficiently powerful database engine and enabling parallel or distributed computing if available.

3. **Privacy**:

   - By default, `minCellCount = 5` is used to align with many data-sharing policies.

   - You may need to mask or remove rows with person counts below that threshold to comply with your organization's privacy rules.

1.5 References

- **The Book of OHDSI**:
  Chapter on Characterization → Cohort Pathways in ATLAS

- **OHDSI Forums**:
  - Cohort Pathways FAQ

  - Reproducing a Treatment Pathway Study JSON Files

1.6 Conclusion

This vignette demonstrated how to run a cohort pathway analysis using **CohortPathways**. You learned how to connect to your database, specify target and event cohorts, execute the analysis, and interpret the resulting data frames. For more complex or large-scale analyses, you may combine this with additional OHDSI tools (such as Hades) or your organization's standard analytics pipeline.

If you have any questions or suggestions, please visit the OHDSI Forums or open an issue on the CohortPathways GitHub repository.