**Understanding circe-be logic through Capr for generating complex cohort definitions**

**Martin Lavallee and Adam Black**

**Background**

ATLAS users creating a cohort definitions to analyze data in the OMOP CDM design high-level constructs like the primary criteria, inclusion rules and cohort exit, populating a “semantic model” for rule-based cohort identification (1). Behind the Atlas user interface there is a complex structure that turns the rule-based logic defined in the ATLAS interface into SQL that generates a person count. This structure, known as circe-be (2) takes the rule-based inputs represented as a json file and converts them to ohdsisql, a parameterized SQL script that is used to query an OMOP database. the HADES package Capr provides an alternative way to populate the sematic model defined by circe-be (3). *Capr* is a programmatic interface to circe-be cohort definitions in R and builds json specifications equivalent to those created by ATLAS that we can use in an OHDSI analytical pipeline. An advantage of *Capr* is that it more directly interfaces with the underlying circe-be structures used to build a cohort definition. In this demonstration we explain the underlying circe-be structures that are critical to creating cohort definitions such as a *concept set* *expression*, *query*, *count*, *group*, and *attribute*. By understanding how these sub-components layer within a cohort definition, we can better understand how to create complex cohort definitions in either ATLAS or *Capr*.

**Methods**

For our example we walk through the eMERGE phenotype for defining a Type 2 Diabetes (T2D) case (4), where previous explanations have also used this example (<https://www.youtube.com/channel/UCZ8nxC-FLq0DMlYRarvJiZg>). This is a complex algorithm with five potential pathways to define a T2D case, as shown in figure 1. To construct this full pathway we need to define the sub-components in the circe-be logic. We use *Capr* as a means of demonstrating each component of the circe-be semantic model and interfacing with these sub-components used to build cohort definitions. We build this cohort definition using the test CMS Synpuf database which includes the latest OMOP vocabulary used to define the logic.

Figure 1: eMERGE T2D Case Algorithm

Diagram

Description automatically generated

**Results**

The demonstration will cover five key components used to create larger aspects of a cohort definition:

1. Concept Set Expression – essentially a code list used to define a clinical event of interest. The expression aspect of this structure adds relational structure to the code set, incorporating descendant logic and adding exceptions to the code list for refined definition
2. Query – allows us to apply our code list to a clinical table in the CDM (i.e. drug exposure, condition occurrence). This allows us to pick a domain and search for patient records
3. Attribute – allows us to filter the query to allow include a subset of persons based on features outside the concept Id. These can be demographic features, numeric values, or dates to name a few.
4. Count – the temporal enumeration of an event of interest relative to an index event. A person timeline is an important consideration for identifying a population. We may only want to include patients who have been exposed to a drug prior to a condition to ensure they have not been previously exposed.
5. Group – a way to bundle multiple counts a deploy logic on top of them. Say we have 3 counts of events that would lead to inclusion, we can specify that all of them must be satisfied, or any or at least n must be satisfied to be included

There is a natural roll-up with these elements that constructs a cohort definition, which will be explained in the demonstration. Next, we explain how to apply these using *Capr*. Understanding these underlying pieces will make it easier to use *Capr* to build cohort definitions and create templating functions. Finally, we construct the T2D case algorithm to show how we can layer the components together.

**Conclusion**

In this demonstration, we show how to use the underlying circe-be components to define complex cohort definitions. We aim to provide greater detail of the semantic model that underlies a cohort definition. The purpose of this explanation is to improve understanding of circe-be in effort to bridge the gap between computational structure and clinical conceptualization of a cohort definition. Bridging this gap will help improve one’s ability to design cohort definitions and to use *Capr* for programmatic creation of cohort definitions.

**References/Citations**

1. Martin Fowler. Domain-Specific Languages. Upper Saddle River, NJ: Addison-Wesley; 2011.

2. Chris Knoll. circe-be [Internet]. 2021. Available from: https://github.com/OHDSI/circe-be

3. Lavallee M. Capr: Cohort definition Application Programming in R. 2022.

4. Pacheco J, Thompson W. Type 2 Diabetes Mellitus. [Internet]. Northwestern University; (PheKB). Available from: https://phekb.org/phenotype/18