Demonstrating Applications of ClinicalCharacteristics 1.0.0 : Improved Optionality for Enumerating Presence of Clinical Events

Casey Tilton, Ajit Londhe, Ron Herrera, Carmen Ng, Katy Sadowski and Martin Lavallee
Boehringer Ingelheim

Background

Previously, we released the OHDSI package ClinicalCharacteristics[1], which is a tool to characterize cohorts of interest via a table-shell approach. We developed this OHDSI R package to first, avoid significant data wrangling needed when using FeatureExtraction[2] and second, to be more explicit and limit characterization to specified concept sets and covariate cohorts as part of the table shell. We have since updated ClinicalCharacteristics by adding improvements to the user interface and underlying object-oriented programming system.

In addition to demonstrating the updated UI of ClinicalCharacteristics, we present methods of characterizing events in a target population. First, we distinguish between characterizing presence of events with or without regard for continuous observation, and second, we distinguish between summarizing covariate cohorts anchored on cohort start date versus cohort era overlap.

Methods

ClinicalCharacteristics allows users to summarize the number of patients who are observed to have presence of a clinical event during a time interval relative to the target cohort index date. Currently ClinicalCharacteristics only supports the use of "earliest event cohort" where there is one cohort entry per patient. For example, we might want to characterize a cohort by condition occurrences that take place 30 to 365 days after index. However, we have found there are different ways of summarizing the presence of an event based on how we choose to apply a patient's observation period and how we evaluate presence of occurrences (i.e., based on the span of patient time versus discrete start dates). Summarizing "presence of a clinical event" can lead to very different results depending on the choices we make in ClinicalCharacteristics.

Any vs Observed Presence

One option for counting presence of clinical event is whether we want to count "any" event or only "observed" events. "Observed" events only count when the event has occurred within the specified time interval and occur during the patient's time in the cohort. "Any" events count when the event has occurred within the specified time interval regardless of the patient's observation period. In both cases the denominator for the percentage is the same — the number of persons in the cohort of interest. Figure 1 shows an example to

differentiate the two methods. If using "any", patients 1, 2, 3, and 4 are counted. If using "observed" we exclude patients 3 and 4 from the count because the event does not occur during the same observed time. We can configure these different options in ClinicalCharacteristics depending on which characterization is best suited for the study. Code chunks 1 and 2 in the Appendix contain examples of how to use the presenceStat option in a ClinicalCharacteristics table shell.

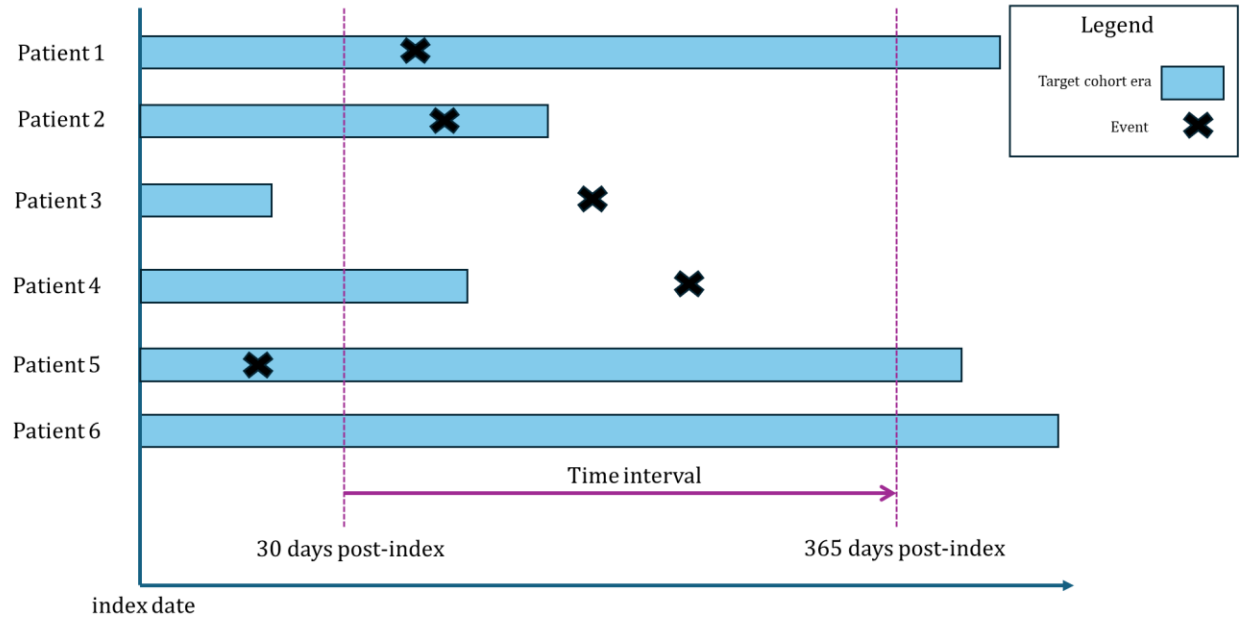
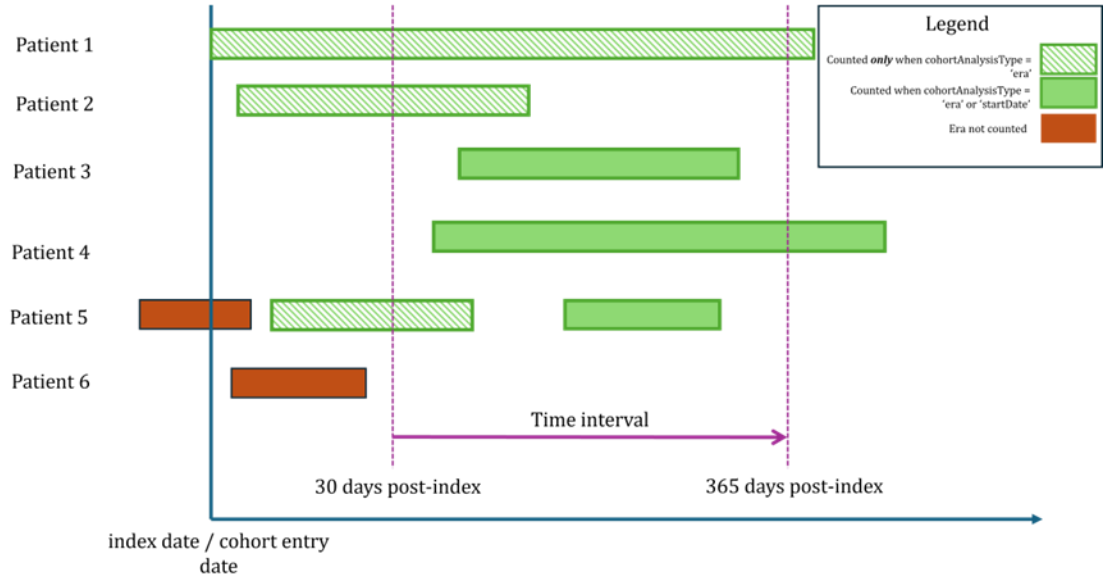


Figure 1: Any vs Observed Presence

Cohort Start Date vs Cohort Era

When characterizing a target cohort by covariate cohorts, we need to determine whether we count an event as present within a time interval using the cohort start date or any covariate cohort era overlap with the interval. Recall that cohorts have an entry and exit date. For example, a drug era cohort starts on the first exposure and persists based on continuous exposure. With this logic, the exposure time is not limited to the start date, but the duration of active exposure, and we want to enumerate the presence of an event as such. With FeatureExtraction, characterizing covariate cohorts can only be done based on cohort era overlap whereas ClinicalCharacteristics provides options for both. Figure 2 demonstrates the difference in enumerating patients based on the cohort era or the start date. Patients 3 and 4 have eras that start after the time interval start date so they would be counted using either analysis type. Patient 5 would qualify under either analysis type because they have an era that overlaps with the time interval start and an era that starts within the time interval. Patient 6 would not be counted by either method because the era

does not overlap with the time interval. Code chunk 3 in the Appendix contains an example of how to set the cohort count type option in a ClinicalCharacteristics table shell.



Results

In this presentation of ClinicalCharacteristics 1.0.0 we demonstrate the new user interface and package design. Further we demonstrate how different approaches for characterizing presence of clinical events can lead to different results. Understanding these differences is crucial in accurately conveying study results to stakeholders. Parameterizing different methods of characterizing event presence has helped us to deliver characterization studies at scale within our organization.

Conclusion

With ClinicalCharacteristics 1.0.0, we aim to provide the OHDSI community with a full suite of methods to characterize cohorts using concept sets and/or cohorts via a table-shell approach. The latest version of ClinicalCharacteristics provides improvements to the user interface and new methods to determine whether patients are eligible to be counted.

Citations

1. Lavallee, Martin, Katy Sadowski, and Ajit Londhe. 2024. ClinicalCharacteristics: Table Shell Approach to OMOP Characterization. https://github.com/OHDSI/ClinicalCharacteristics.

2. Schuemie, Martijn, Marc Suchard, Patrick Ryan, Jenna Reps, Anthony Sena, and Ger Inberg. 2024. *FeatureExtraction: Generating Features for a Cohort*. https://ohdsi.github.io/FeatureExtraction.

Appendix

Code example 1: set statistic = anyPresenceStat() within createConceptSetLineItem()

```
library(ClinicalCharacteristics)

# Define Execution Settings
executionSettings <- createExecutionSettings(
  connectionDetails = connectionDetails,
  cdmDatabaseSchema = "omop_schema",
  workDatabaseSchema = "my_schema",
  tempEmulationSchema = "my_schema",
  cohortTable = "my_cohorts",
  cdmSourceName = "my_omop"
)

ckd_cs <- Capr::cs(Capr::descendants(46271022), name = "ckd")
tw1 <- timeInterval(lb = -365, rb = -1)

ts <- createTableShell(
  title = "Demo 1",
  targetCohorts = list(
    createCohortInfo(id = 1, name = "Type 2 Diabetes")
  ),
  lineItems = lineItems(
    createConceptSetLineItem(
      sectionLabel = "CKD: Baseline",
      domain = "condition_occurrence",
      statistic = anyPresenceStat(),
      conceptSet = ckd_cs,
      timeInterval = tw1
    )))
```

Code example 2: set statistic = observedPresenceStat() within createConceptSetLineItem()

```
ts <- createTableShell(
  title = "Demo 1",
  targetCohorts = list(
    createCohortInfo(id = 1, name = "Type 2 Diabetes")
  ),
  lineItems = lineItems(
    createConceptSetLineItem(
```

```
    sectionLabel = "CKD: Baseline",
    domain = "condition_occurrence",
    statistic = observedPresenceStat(),
    conceptSet = ckd_cs,
    timeInterval = tw1
  )))
```

Code example 3: set cohortAnalysisType to “era” or “startDate” within a BuildOptions class object

```
buildOptionsEra = BuildOptions$new(codesetTempTable = "#codeset",
                                   sourceCodesetTempTable = "#source_codeset",
                                   timeWindowTempTable = "#time_windows",
                                   targetCohortTempTable = "#target_cohorts",
                                   tsMetaTempTable = "#ts_meta",
                                   conceptSetOccurrenceTempTable = "#concept_set_occ",
                                   cohortOccurrenceTempTable = "#cohort_occ",
                                   patientLevelDataTempTable = "#patient_data",
                                   patientLevelTableShellTempTable = "#pat_ts_tab",
                                   categoricalSummaryTempTable = "#categorical_table",
                                   continuousSummaryTempTable = "#continuous_table",
                                   cohortAnalysisType = “era”
                                   ) {

buildOptionsStartDate = BuildOptions$new(codesetTempTable = "#codeset",
                                   sourceCodesetTempTable = "#source_codeset",
                                   timeWindowTempTable = "#time_windows",
                                   targetCohortTempTable = "#target_cohorts",
                                   tsMetaTempTable = "#ts_meta",
                                   conceptSetOccurrenceTempTable = "#concept_set_occ",
                                   cohortOccurrenceTempTable = "#cohort_occ",
                                   patientLevelDataTempTable = "#patient_data",
                                   patientLevelTableShellTempTable = "#pat_ts_tab",
                                   categoricalSummaryTempTable = "#categorical_table",
                                   continuousSummaryTempTable = "#continuous_table",
                                   cohortAnalysisType = “startDate”
                                   ) {


res <- generateTableShell(tableShell = ts, executionSettings = executionSettings,
buildOptions = buildOptionsEra)
```