Introduction to Strategus

Anthony G. Sena

2024-06-04

Contents

1 Introduction to Strategus

The `Strategus` package is a new approach for coordinating and executing analytics using HADES modules. The goal is to have OHDSI network sites install `Strategus` and exchange an analysis specification in JSON format to execute a network study. The analysis specification will capture all of the design choices that pertain to the methods used in a given study. The analysis specification format aims to allow for combining different HADES modules together as a pipeline to execute a study. In addition, the analysis specification makes use of specific HADES module versions to clearly state which modules are required to run the study with the aim to make the execution and results reproducible.

1.1 Using Strategus

The high-level steps in using `Strategus` consists of the following:

1. Create the analysis specification for the study. This will include things like picking the cohorts for the study, the analyses to perform (i.e. Cohort Diagnostics, Comparative Cohort Study, etc) and the choices for each analysis. See here for more details.
2. Create the execution settings that specify how to connect to the OMOP CDM in your environment. See here for more details.
3. Execute the study using `Strategus` which will require both the analysis specification and the execution settings in order to run the study and extract the results. See here for more details.
4. Upload the results and use Shiny to view the results *Coming Soon*.

1.2 What is a HADES module?

A HADES module aims to standardize the input and output produced by a HADES package. It also self-describes the R dependencies it needs to run properly via `Strategus`.

From a technical perspective, a HADES module is a lightweight R script uses the `Strategus` JSON analysis specification to call one or more HADES packages to produce a standardized set of results. The HADES module contains an renv lock file which enumerates the module's R package dependencies. Taken together, the module provides a reproducible way to execute a step in an analysis by ensuring the proper dependencies are available along with the code that was used to execute the analysis.

For more information on how to create modules, see here