Strategus: Marching towards transparent, reproducible research

Anthony G. Sena[1], Christopher Knoll[1], James Gilbert[1], Jenna Reps[1], Frank DeFalco[1], Clair Blacketer[1], Anthony Molinaro[1], Joshua Ide[1], Patrick Ryan[1], Martijn Schuemie[1]

[1]Janssen Research & Development, LLC, Titusville, NJ, United States

Background

Observational Health Data Science and Informatics (OHDSI) best practices for network studies recommends creating a study package "to generate a completely traceable and reproducible process documented in the form of computer code." [1] This study package largely comes in the form of an R[2] package that utilizes the HADES[3] libraries to fully encapsulate the study design and steps to perform the required analytics. For every OHDSI network study, an organization must follow these steps: download the study package, configure and execute the package in their environment and share results with the network study coordinator. These steps are not always straightforward for different OHDSI network sites, often requiring Information Technology resources for the configuration and execution of each study package.

In this software demonstration, we will showcase Strategus which aims to change our approach to performing OHDSI network studies. One of the goals of Strategus is to have a single R infrastructure available to execute a network study based on a JavaScript Object Notation (JSON) document which fully specifies the inputs and analytics to execute.

Methods

Strategus[4] takes as input two documents to execute a study: the analytics specification and the execution settings. The analytics specification fully describes the study design choices, such as the cohorts to use in the analysis and which analytics to perform. The execution settings are site specific and are used to specify settings such as the connection to the OMOP CDM database and where to store results files. Taking the analytics specification and execution settings together allows for the full execution of the study design by Strategus.

Individual analytics (i.e., characterization, population-level estimation, patient level prediction) that are specified in the analytics specification are referred to as "modules" that run in Strategus. A module is designed as an R project that uses renv[5] to enumerate its R package dependencies. A module provides a wrapper around one or more HADES packages and is responsible for taking the JSON input specification, performing the analytics and writing the output files. Additionally, a module can enforce best practices around performing study diagnostics in conjunction with producing the analytical results. Modules form the building blocks for conducting a portion of the study analytics and are reusable across different studies. Furthermore, the architecture of Strategus modules allow for development of new modules that can "plug in" to the overall Strategus execution pipeline of tasks.

Strategus' code base also makes use of the 'targets' R package[6] to allow for creating a pipeline of tasks, based on the analytics specification, to potentially parallelize operations across a compute cluster if such infrastructure is available.

Results

Figure 1 shows the format of an example input analytic specification and execution settings for use in

Strategus. These settings are then used to download the modules specified from GitHub, install all dependencies as detailed in the renv.lock file for each module and then create an execution environment to perform the steps as detailed in the *moduleSpecification* section.

Analytic Specification


```
{
  "sharedResources": [
    {
      "cohortDefinitions": [
        {
          "cohortId": "1",
          "cohortName": "celecoxib",
          "cohortDefinition": "{\n  \"ConceptSets\": [\n    {\n      \"id\": 0,\n      \"r
        },
        {…
        },
        {…
        }
      ],
      "attr_class": ["CohortDefinitionSharedResources", "SharedResources"]
    }
  ],
  "moduleSpecifications": [
    {
      "module": "CohortGeneratorModule",
      "version": "0.0.5",
      "remoteRepo": "github.com",
      "remoteUsername": "ohdsi",
      "settings": {
        "incremental": true,
        "generateStats": true
      },
      "attr_class": ["CohortGeneratorModuleSpecifications", "ModuleSpecifications"]
    },
    {
      "module": "CohortDiagnosticsModule",
      "version": "0.0.2",
      "remoteRepo": "github.com",
      "remoteUsername": "ohdsi",
      "settings": {
        "runInclusionStatistics": true,
        "runIncludedSourceConcepts": true,
        "runOrphanConcepts": true,
        "runTimeSeries": false,
        "runVisitContext": true,
        "runBreakdownIndexEvents": true,
        "runIncidenceRate": true,
        "runCohortRelationship": true,
        "runTemporalCohortCharacterization": true,
        "temporalCovariateSettings": {…
        },
        "incremental": false
      },
      "attr_class": ["CohortDiagnosticsModuleSpecifications", "ModuleSpecifications"]
```


Execution Settings


```
{
  "connectionDetailsReference": "Eunomia",
  "workDatabaseSchema": "main",
  "cdmDatabaseSchema": "main",
  "cohortTableNames": {
    "cohortTable": "strategus_test",
    "cohortInclusionTable": "strategus_test_inclusion",
    "cohortInclusionResultTable": "strategus_test_inclusion_result",
    "cohortInclusionStatsTable": "strategus_test_inclusion_stats",
    "cohortSummaryStatsTable": "strategus_test_summary_stats",
    "cohortCensorStatsTable": "strategus_test_censor_stats"
  },
  "workFolder": "c:/temp/strategusWork",
  "resultsFolder": "c:/temp/strategusOutput",
  "minCellCount": 5,
  "attr_class": "ExecutionSettings"
}
```


Figure 1. Input analytic specifications and execution settings JSON documents for Strategus.

Strategus then constructs a targets pipeline to execute the modules based on the analytic speciation. Figure 2 provides a visual example of a target pipeline.

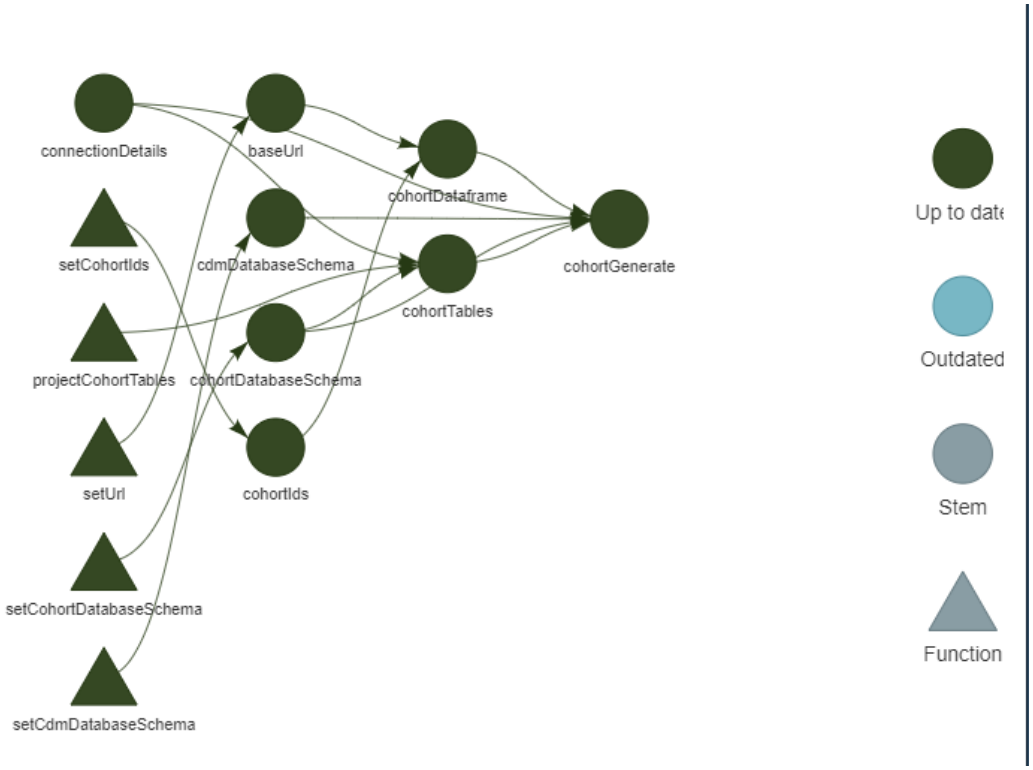

Figure 2. Example targets pipeline

The results of the pipeline are then written to the file system. Each module will produce a set of results in human readable format, generally comma-separated value (CSV) files. This follows OHDSI best practices for conducting network studies as it allows for each site to review results before submitting to the study coordinator.

Our software demonstration will provide details on the following topics:

- How to create input specifications for use in Strategus.
- How to construct an analytic module for use in Strategus.
- Demonstrate how we can execute a Strategus targets pipeline

Conclusion

In this software demonstration, we will demonstrate Strategus as a new vision for OHDSI network studies that aims to simplify the R infrastructure requirements for network sites. In this new paradigm, network studies will specify their study design in a JSON document that fully encapsulates all the inputs and choices required for performing the analytics. Reducing the burden of deploying R packages for each study should enable more rapid deployment and execution of studies across the OHDSI network. As a result, we should be able to perform more pro-active safety studies in a transparent and reproducible manner consistent with OHDSI best practices.

References/Citations

1. OHDSI. Book Of OHDSI Chapter 19 – Study Steps. Available from: https://ohdsi.github.io/TheBookOfOhdsi/StudySteps.html
2. R Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Available from: https://www.R-project.org
3. Health Analytics Data-to-Evidence Suite (HADES). Available from: https://ohdsi.github.io/Hades/
4. OHDSI. Strategus R package: https://github.com/OHDSI/Strategus/
5. Ushey K (2022). renv: Project Environments. Available from: https://rstudio.github.io/renv/
6. Landau WM (2021). "The targets R package: a dynamic Make-like function-oriented pipeline toolkit for reproducibility and high-performance computing." Journal of Open Source Software, 6(57), 2959. https://doi.org/10.21105/joss.02959