

Execute Strategus

Anthony G. Sena

2024-01-26

Contents

1	Creating execution settings	1
1.1	Storing the connection details	1
1.2	Creating an execution settings object	2
2	Executing the study	2
2.1	Specifying the instantiated modules folder	2
2.2	Running the study	3

A Strategus study is defined by *analysis specifications*. These specifications describe which modules to run, with which settings. The ‘Creating Analysis Specification’ vignette describes how to create analysis specifications. In this vignette, we demonstrate how to run a study once it is specified.

1 Creating execution settings

In addition to analysis specifications, Strategus also requires *execution settings*. The execution settings specify how the study should be executed in a specific environment, for example how to connect to a database, and what local folders to use. Many Strategus studies run against data in the OMOP Common Data Model (CDM), and in this vignette we focus on this type of studies. (Other studies, such as meta-analyses, may run against results data instead). In this example, we will make use of the Eunomia data set which is an OMOP CDM with simulated data used for example purposes. When running a study against your own CDM data, you will need to specify the database connection details for your environment. Execution settings for studies against the CDM can be created using `createCdmExecutionSettings()`.

Note that Strategus uses keyring to store the connection information securely, so that sensitive information is not stored directly in the `executionSettings` variable.

1.1 Storing the connection details

In this example, we first create a `connectionDetails` reference for Eunomia. In your environment, the `connectionDetails` would be specific to your OMOP CDM connection.

```
library(Strategus)
folder <- "c:/strategus"
```

```
# Note: specifying the `databaseFile` so the Eunomia database is permanent, not temporary:
connectionDetails <- Eunomia::getEunomiaConnectionDetails(
  databaseFile = file.path(outputFolder, "cdm.sqlite")
)
```

Next we will use `Strategus` to store the connection details and provide a `connectionDetailsReference` that `Strategus` will use to look up the connection details.

```
storeConnectionDetails(
  connectionDetails = connectionDetails,
  connectionDetailsReference = "eunomia"
)
```

Note that we only need to store connection details once. We can re-use these details in all future studies.

1.2 Creating an execution settings object

Next, we will use `Strategus` to create the CDM execution settings. The `connectionDetailsReference` refers to the connection details we stored earlier:

```
executionSettings <- createCdmExecutionSettings(
  connectionDetailsReference = "eunomia",
  workDatabaseSchema = "main",
  cdmDatabaseSchema = "main",
  cohortTableNames = CohortGenerator::getCohortTableNames(),
  workFolder = file.path(outputFolder, "work_folder"),
  resultsFolder = file.path(outputFolder, "results_folder"),
  minCellCount = 5
)
```

Finally, we can write out the execution settings to the file system to capture this information.

```
ParallelLogger::saveSettingsToJson(
  object = executionSettings,
  file.path(outputFolder, "eunomiaExecutionSettings.json")
)
```

2 Executing the study

2.1 Specifying the instantiated modules folder

`Strategus` studies use modules to perform the analytics, and each module has a version. Different studies can use the same modules, but a single study can only use one version of a particular module. These modules need to be instantiated before they can be executed, a sometimes time-consuming task. We must specify a global location where these modules will be instantiated so that, once a module is instantiated, it can be used in all future studies:

```
Sys.setenv("INSTANTIATED_MODULES_FOLDER" = "c:/strategus/modules")
```

We recommend adding this environmental variable to your `.renviron` file, so it is always set.

2.2 Running the study

For this study, we will use analysis specifications created elsewhere, and the execution settings we created earlier:

```
analysisSpecifications <- ParallelLogger::loadSettingsFromJson(  
  fileName = system.file("testdata/analysisSpecification.json",  
    package = "Strategus"  
  )  
)  
  
executionSettings <- ParallelLogger::loadSettingsFromJson(  
  fileName = file.path(outputFolder, "eunomiaExecutionSettings.json")  
)
```

And finally we execute the study:

```
execute(  
  analysisSpecifications = analysisSpecifications,  
  executionSettings = executionSettings,  
  executionScriptFolder = file.path(outputFolder, "script_folder")  
)
```

This will first instantiate all the modules if they haven't already been instantiated, and will then execute each module in sequence according to the analysis specifications. The results will appear in subfolders of the 'results_folder', as specified in the execution settings.