Sisyphus Challenge Week 6: Strategus execution

Jenna/Jack
Sena/Chungsoo
Engineering open science systems that build trust into the real-world evidence generation and dissemination process

'System' required elements:
- Required phenotypes
- Analysis specifications
- Decision thresholds

Distributed data network, standardized to common data model

Network coordination

Data quality evaluation
Research question \(\rightarrow\) Database diagnostics \(\rightarrow\) Pass

Phenotype development and evaluation
Cohort definitions \(\rightarrow\) Cohort diagnostics \(\rightarrow\) Pass

Analysis reliability evaluation
Analysis design choices \(\rightarrow\) Study diagnostics \(\rightarrow\) Fail

Week 6-8: Analysis execution, study diagnostics, evidence synthesis

Final unblinded results

Interface for exploration

Required elements:

- Required phenotypes
- Analysis specifications
- Decision thresholds

Distributed data network, standardized to common data model

Network coordination
Standardized analyses currently available within Strategus pipeline

- **Characterization**
  - Cohort diagnostics
  - Cohort features
  - Incidence rates
  - Time-to-event
  - Dechallenge / rechallenge

- **Patient-level prediction**

- **Population-level effect estimation**
  - Comparative cohort
  - Self-controlled case-series (SCCS)
OHDSI software for network studies

Strategus

Pros
- Standardized representation of study inputs as JSON
- Modular composition of HADES analytics packages for study design
- Handles R environment configuration using renv

Reality
- Strategus is still in development
Strategus Modules

What is inside of a module?
How is it different from a HADES package?
Strategus Modules

**Strategus Module** is an R Project that acts as a lightweight wrapper around one or more HADES packages. It has the following responsibilities:

- Declaring an renv.lock file for all R dependencies
- Providing functions for defining the module settings to store in the analysis specification JSON
- Implement an “execute” function to perform the analytical tasks based on module settings JSON using HADES packages
- Writing results as CSV files for sharing across the OHDSI network
Strategus Modules

- Strategus modules can be combined to accommodate various study designs.
Prerequisites to execute Strategus study

- **Database with OMOP CDM data**: convert your data to OMOP CDM and set up database server
- **Software**: Install R/R Studio + Java + setup GitHub personal access token (see HADES instructions: [https://ohdsi.github.io/Hades/rSetup.html](https://ohdsi.github.io/Hades/rSetup.html))
Strategus execution: 5 Steps in < 15 minutes

1. Download project
2. Configure local settings
3. Execute
4. Review CSV files
5. Share Results
Download Project

1. Go to study GitHub repo:  https://github.com/ohdsi-studies/AntiVegfKidneyFailure
2. Download zipped folder
3. Unzip to local folder
4. Open folder and double click 'AntiVegfKidneyFailure.Rproj'
Configure

After opening AntiVegfKidneyFailure.Rproj you should see:
Configure local settings

Click on KeyringSetup.R to open the file in R Studio
Configure

1. Install keyring
2. Add strategus password to renviron
3. Configure database settings

---

```r
# Install keyring - one time operation --------
install.packages("keyring")

# Set keyring password by adding STRATEGUS_KEYRING_PASSWORD='sos' to renviron
if (Sys.getenv("STRATEGUS_KEYRING_PASSWORD") == "") {
    # set keyring password by adding STRATEGUS_KEYRING_PASSWORD='sos' to renviron
    usethis::edit_r_environ()
    # then add STRATEGUS_KEYRING_PASSWORD='sos', save and close
    # Restart your R Session to confirm it worked
    stop("Please add STRATEGUS_KEYRING_PASSWORD='sos' to your .Renviron file via usethis::edit_r_environ() as instructed.")
}

# Provide your environment specific values ------
dbms <- "redshift"
connectionString <- "jdbc:redshift://your.server.goes.here:5439/your_cdm_database"
username <- "username-goes-here"
password = "password-goes-here"

# Run the rest to setup keyring ----------
```
• Open 'StrategusCodeToRun.R'
1. Install Strategus

2. Specify result output location and database settings

3. Specify connection details
Review CSV results

The 'strategusOutput' folder contains the results for each module that was executed.
Inside strategusOutput folder
Each module folder contains csv files.
To share csv results:

Add database reference
Add location of results

Get keyFileName and userName from study coordinator
5 Step Summary

• **Download:** Download zip in github page (https://github.com/ohdsi-studies/AntiVegfKidneyFailure), unzip folder in local directory and open R project (AntiVegfKidneyFailure.Rproj).

• **Configure:** Open KeyringSetup.R to configure database connection settings and keyring password using keyring

• **Execute:** Open StrategusCodeToRun.R to edit local output directory and database parameters and then run study using those settings

• **Review:** explore csv files in StrategusOutput and/or launch shiny

• **Share:** open ShareResults.R to export result files to OHDSI
Viewing shiny app

• First you need to upload the results to a database, we recommend sqlite for local viewing.

• To upload results into local sqlite database open the file "StrategusResultsUpload.R"
You only need to specify two inputs then run the script to insert the results into a local sqlite database.

```r
strategies

---

AntivegfKidneyFailure / StrategusResultsUpload.R

---

anthonysema Updates for upload and shiny

---

1 contributor

---

163 lines (151 sloc) 6.4 KB

---

1 #== START OF INPUTS ==#

2

3 connectionDetailsReference <- "Jmdc"

4 outputLocation <- 'D:/git/anthonysema/AntivegfKidneyFailure'

---

"#### END OF INPUTS =="
Your data are now ready for the shiny app

- Launch shiny app open the file "app.R"

```r
# View Strategus results in the results database
remotes::install_github("ohdsi/ShinyAppBuilder", ref = "develop")
remotes::install_github("ohdsi/OhdsiShinyModules", ref = "develop")
library(dplyr)
library(ShinyAppBuilder)
library(markdown)

#### START OF INPUTS

connectionDetailsReference <- "jmdc"
outputLocation <- '"D:/git/anthonysena/AntiVegfKidneyFailure'`

#### END OF INPUTS
```

Only two inputs are required (the same values used when creating the local sqlite database on slide 22)
Shiny App

Let's explore...

Table of contents

1. Introduction
2. How to use the viewer
3. Analysis types
   1. Characterization
   2. Population-level effect estimation
   3. Patient-level prediction

Introduction

This is an interactive shiny app for exploring standardized outputs for OHDSI analyses including:

- characterization (descriptive studies)
- population-level effect estimation (causal inference)
- patient-level prediction (inference)

Full details of all the analysis tools can be found on the HADES website
Characterization

Explore incidence rates, event timing, compare those target patients with and without the outcome during the time at risk...
Cohort Method

View diagnostics and cohort method results (if they passed diagnostics), explore the analyses...
Self-Controlled Case Series

View diagnostics and SCCS method results (if they passed diagnostics), explore the analyses...
PatientLevelPrediction

View how feasible it was to predict the outcomes within the target populations during the time at risk...

<table>
<thead>
<tr>
<th>Design</th>
<th>Model Type</th>
<th>Target Pop</th>
<th>Outcome</th>
<th>TAR</th>
<th>min AUROC</th>
<th>mean AUROC</th>
<th>max AUROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>logistic</td>
<td>Cohort: 1782483001</td>
<td>[SOS] End-stage renal disease</td>
<td>(cohort start + 1) - (cohort start + 365)</td>
<td>0.925</td>
<td>0.925</td>
<td>0.925</td>
</tr>
<tr>
<td>2</td>
<td>logistic</td>
<td>Cohort: 1782483002</td>
<td>[SOS] End-stage renal disease</td>
<td>(cohort start + 1) - (cohort start + 365)</td>
<td>0.930</td>
<td>0.930</td>
<td>0.930</td>
</tr>
</tbody>
</table>
We have a central shiny app

- Results will be uploaded to https://data.ohdsi.org/AntiVegfKidneyFailure/
- Check out that app throughout the week as more results are added...