HADES

Health Analytics Data-to-Evidence Suite

Open-Source Software for Observational Research

Martijn Schuemie
Epidemiology department, Johnson & Johnson
Department of Biostatistics, UCLA

What is HADES?

Health Analytics Data to Evidence Suite (HADES)

- Collection of 35 R packages used in almost every OHDSI study
  - OHDSI is an international collaborative for observational research
- Run directly against data in the OMOP Common Data Model (CDM)
  - Health care insurance claims
  - Electronic health records
- Perform observational analyses
  - Characterization
  - Causal effect estimation
  - Patient-level prediction

https://ohdsi.github.io/Hades/

HADES design principles

- Promote **open science** through open source
- Execute directly against the **OMOP CDM**
- Implement **best practices** as informed by methods research
- Provide **high quality software** (documented, maintained, tested, validated)
- Facilitate **large-scale analytics**, answering many questions at once
- Support **big data**, covering hundreds of millions of lives
- Enable **federated analyses**
- Run **across a wide variety of technical infrastructures**

HADES – supporting packages

- **DatabaseConnector + SqlRender**: write code once, run on all supported platforms (SQL Server, Oracle, Postgres, RedShift, BigQuery, DataBricks, Snowflake)
- **Andromeda**: Work with data objects too big to fit in memory
- **ParallelLogger**: extensive logging to facilitate remote debugging
- **Cyclops**: fit very large regression models (logistic, Poisson, Cox)
- **DataQualityDashboard**: evaluate data quality

HADES – cohort packages

We define a cohort as a set of persons who satisfy one or more inclusion criteria for a duration of time.

- Exposure cohorts (e.g. people exposed to warfarin)
- Outcome cohorts (e.g. people experiencing GI bleeding)
- Cohorts of special interest (e.g. pregnant women)

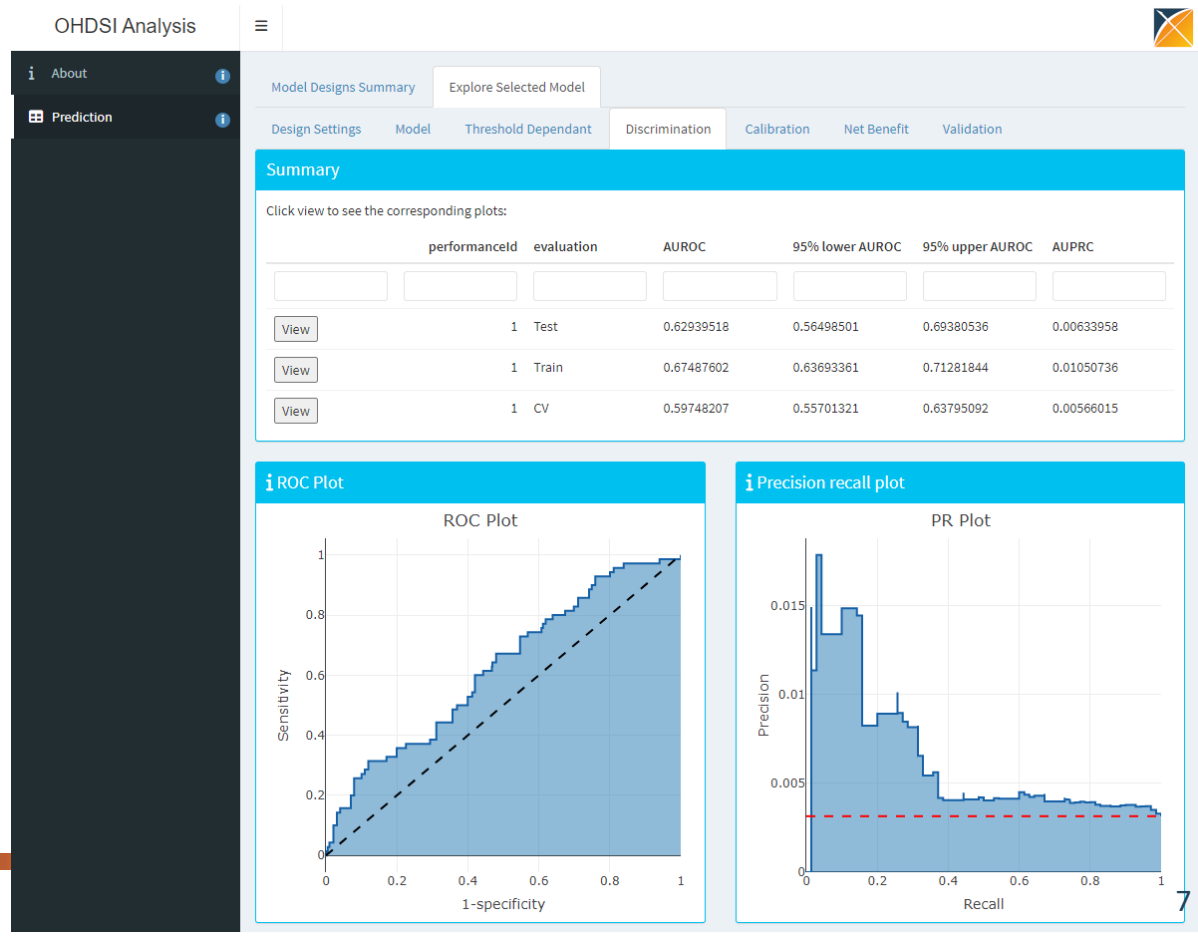
- **Capr**: Define cohorts using complex logic
- **PhenotypeLibrary**: for storing OHDSI-approved cohort definitions
- **CirceR**: for turning cohort definitions into SQL or human-readable text
- **CohortGenerator**: for instantiating cohorts in a database
- **CohortDiagnostics** and **PheValuator**: for evaluating cohorts

HADES – analytics packages

- **PatientLevelPrediction**: develop and evaluate prediction models
- **CohortMethod** and **SelfControlledCaseSeries**: estimate causal effects
- **EmpiricalCalibration**: Calibrate causal effect estimates based on negative controls
- **EvidenceSynthesis**: Combine causal effect estimates across databases without sharing patient-level data.

HADES analytics output

- R objects possibly containing patient-level data
- CSV files / database tables for sharing
- Shiny apps



Publications using HADES

- 38 peer-reviewed clinical research papers
- 29 methods research papers

And now…

code for performing a comparative safety study using the CohortMethod package

Download all required data from server

```
covSettings <- createDefaultCovariateSettings(
    excludedCovariateConceptIds = c(diclofenacConceptId,
                                    celecoxibConceptId),
    addDescendantsToExclude = TRUE
)
cohortMethodData <- getDbCohortMethodData(
    connectionDetails = connectionDetails,
    cdmDatabaseSchema = cdmDatabaseSchema,
    targetId = 1,
    comparatorId = 2,
    outcomeIds = 77,
    exposureDatabaseSchema = cohortDatabaseSchema,
    exposureTable = cohortTable,
    outcomeDatabaseSchema = cohortDatabaseSchema,
    outcomeTable = cohortTable,
    covariateSettings = covSettings
)
```

Create covariates for all conditions, drugs, procedures, etc.

Assuming exposure and outcome cohorts have already been created

Estimate causal effect

```
studyPop <- createStudyPopulation(
      cohortMethodData = cohortMethodData,
      outcomeId = 77,
      removeDuplicateSubjects = "keep first",
      removeSubjectsWithPriorOutcome = TRUE,
      minDaysAtRisk = 1,
      riskWindowStart = 0,
      startAnchor = "cohort start",
      riskWindowEnd = 30,
      endAnchor = "cohort end "
)
ps <- createPs(cohortMethodData = cohortMethodData,
                  population = studyPop)
matchedPop <- matchOnPs(ps, maxRatio = 1)
outcomeModel <- fitOutcomeModel(matchedPop, modelType = "cox")
```

Interested in the first 30 days after exposure

Fit a propensity model using LASSO

1-1 PS match Cox regression

Diagnostics

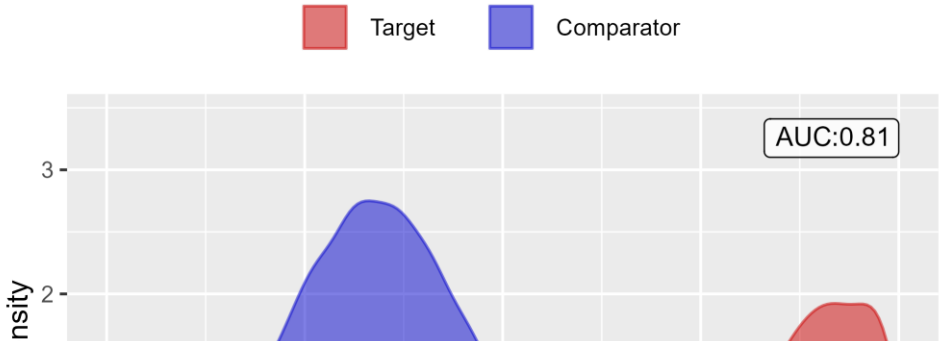
```
plotPs(ps, showAucLabel = TRUE)
```

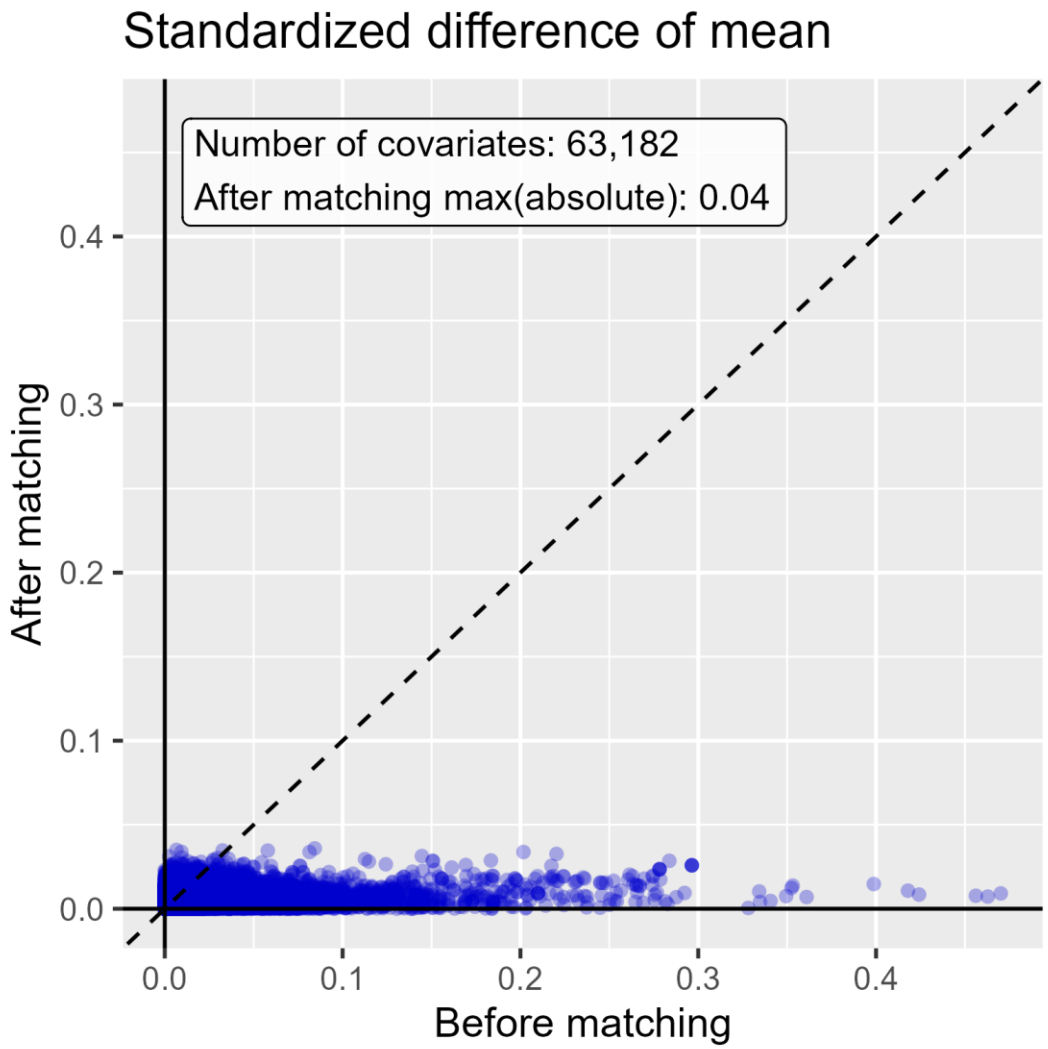
Diagnostics

```
plotPs(ps, showAucLabel = TRUE)

balance <- computeCovariateBalance(
    matchedPop,
    cohortMethodData
)

plotCovariateBalanceScatterPlot(
    balance,
    showCovariateCountLabel = TRUE,
    showMaxLabel = TRUE
)
```





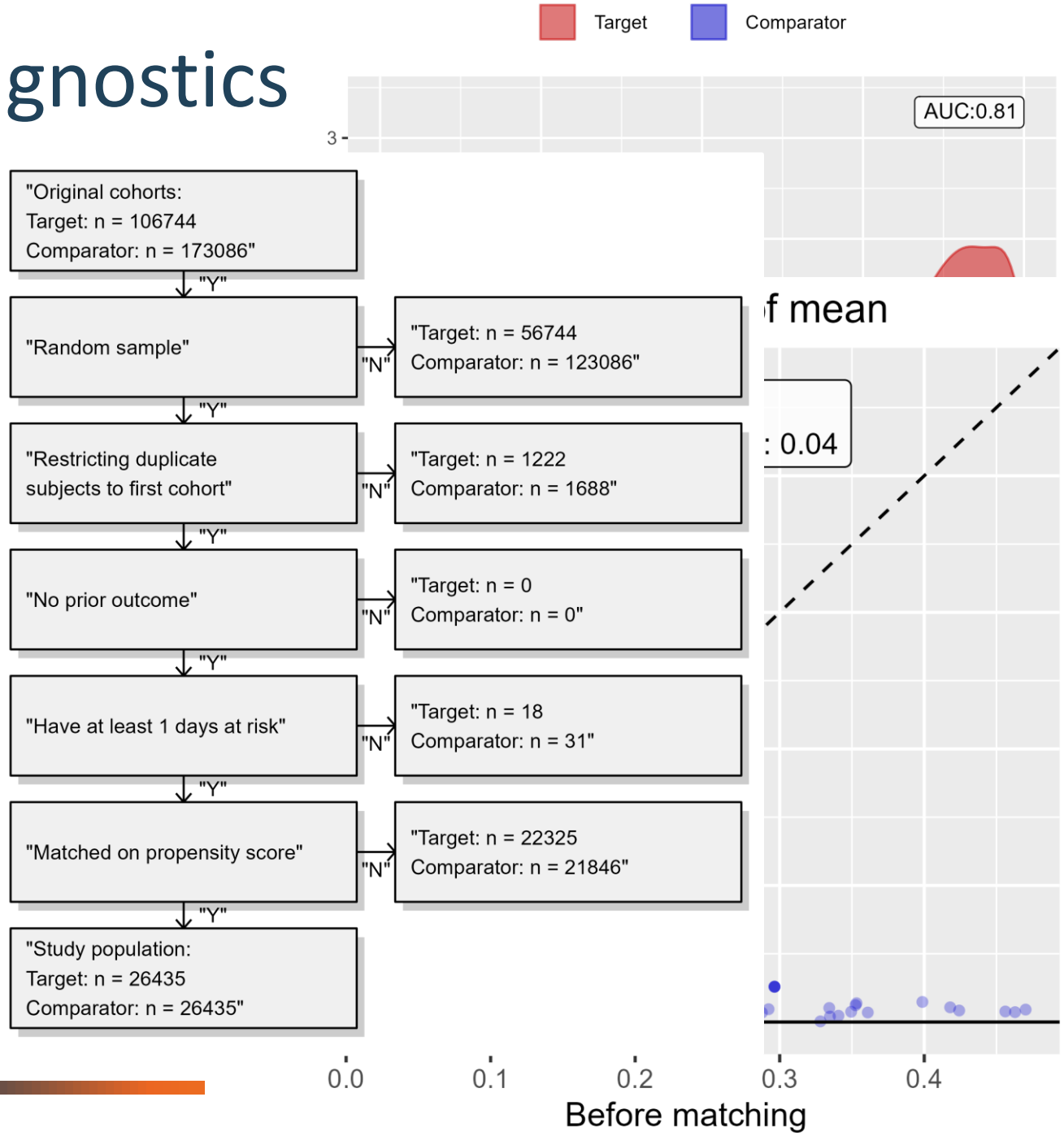
Diagnostics

```
plotPs(ps, showAucLabel = TRUE)

balance <- computeCovariateBalance(
    matchedPop,
    cohortMethodData
)

plotCovariateBalanceScatterPlot(
    balance,
    showCovariateCountLabel = TRUE,
    showMaxLabel = TRUE
)

drawAttritionDiagram(outcomeModel)
```



Diagnostics

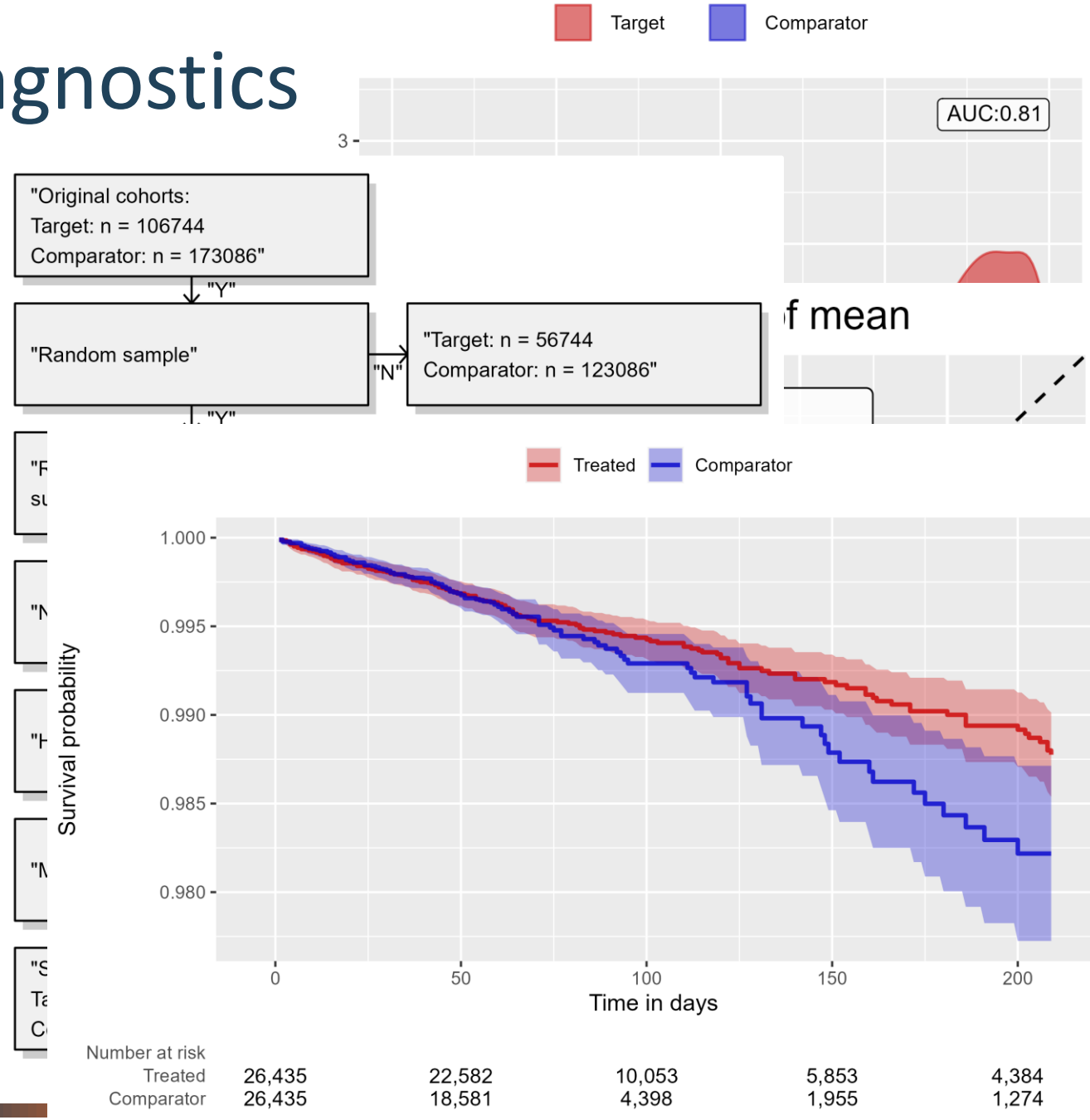
```
plotPs(ps, showAucLabel = TRUE)

balance <- computeCovariateBalance(
    matchedPop,
    cohortMethodData
)

plotCovariateBalanceScatterPlot(
    balance,
    showCovariateCountLabel = TRUE,
    showMaxLabel = TRUE
)

drawAttritionDiagram(outcomeModel)

plotKaplanMeier(outcomeModel)
```



Conclusions

- HADES is a suite of R packages for analyzing observational healthcare data
- Thanks to the OMOP Common Data Model, HADES runs on a wide variety of data sources across the world
- Open source, to promote open science (all analytics code can be shared as part of publication)
- Supports federated networks, where data stay locally, and results are shared


OHDSI
OBSERVATIONAL HEALTH DATA SCIENCES AND INFORMATICS

Thank you!

```
library(DatabaseConnector)
Library(keyring)
connectionDetails <- createConnectionDetails(
        dbms = "redshift",
        connectionString = key_get("redShiftConnectionString"),
        user = key_get("redShiftUserName"),
        password = key_get("redShiftPassword")
)
cdmDatabaseSchema <- "cdm_truven_mdcr_v2322"
cohortDatabaseSchema <- "scratch_mschuemi"
cohortTable  <- "cm_vignette"
```

```
library(Capr)
library(CirceR)
osteoArthritisOfKneeConceptId <- 4079750
celecoxibConceptId <- 1118084
diclofenacConceptId <- 1124300
osteoArthritisOfKnee <- cs(descendants(osteoArthritisOfKneeConceptId),
                           name = "Osteoarthritis of knee")
attrition = attrition("prior osteoarthritis of knee" = withAll(
    atLeast(1, condition(osteoArthritisOfKnee), duringInterval(eventStarts(-Inf, 0))))
)
celecoxibCs <- cs(descendants(celecoxibConceptId, name = "Celecoxib")
diclofenacCs  <- cs(descendants(diclofenacConceptId), name = "Diclofenac")
celecoxibCohort <- cohort(entry = entry(drug(celecoxibCs, firstOccurrence()),
                                        observationWindow = continuousObservation(priorDays = 365)),
    attrition = attrition,
    exit = exit(endStrategy = drugExit(celecoxibCs, persistenceWindow = 30, surveillanceWindow = 0)))
diclofenacCohort <- cohort(entry = entry(drug(diclofenacCs, firstOccurrence()),
                                         observationWindow = continuousObservation(priorDays = 365)),
    attrition = attrition,
    exit = exit(endStrategy = drugExit(diclofenacCs, persistenceWindow = 30, surveillanceWindow = 0)))
```

```
exposureCohorts <- tibble(
    cohortId = c(1, 2),
    cohortName = c("Celecoxib", "Diclofenac"),
    json = c(as.json(celecoxibCohort),
             as.json(diclofenacCohort))
)
exposureCohorts$sql <- sapply(exposureCohorts$json,
                              buildCohortQuery,
                              options = createGenerateOptions())

library(PhenotypeLibrary)
outcomeCohorts <- getPlCohortDefinitionSet(77) # GI bleed
```

```
library(CohortGenerator)
allCohorts <- bind_rows(outcomeCohorts,
                        exposureCohorts)
cohortTableNames <- getCohortTableNames(cohortTable = cohortTable)
createCohortTables(
    connectionDetails = connectionDetails,
    cohortDatabaseSchema = cohortDatabaseSchema,
    cohortTableNames = cohortTableNames)
generateCohortSet(
    connectionDetails = connectionDetails,
    cdmDatabaseSchema = cdmDatabaseSchema,
    cohortDatabaseSchema = cohortDatabaseSchema,
    cohortTableNames = cohortTableNames,
    cohortDefinitionSet = allCohorts
)
```