



The CohortMethod R package & Review of code generated by ATLAS



CohortMethod package

OHDSI / CohortMethod

An R package for performing new-user cohort studies in an observational database in the OMOP Common Data Model.

445 commits 9 branches 10 releases 7 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

msuchard trying repo slug

- .settings Mysterious new EOL problem
- R Fixed typo causing bound
- data The treatment variable
- extras Fixed typo causing bound
- inst Reran multi-analysis vign
- man Power calculations now a
- src Now using default select
- standalone/src Fixed end of line charact
- tests Added option to choose
- vignettes Minor edits to vignettes
- .Rbuildignore enable drat
- .cproject Mysterious new EOL problem

GitHub repository:

- Installation instructions
- Package manual
- Vignettes

<https://github.com/OHDSI/CohortMethod>



CohortMethod package

What is it?

- Library of R commands
- Few lines of code → full cohort study
- Flexible (if you know R)
- ‘Talks’ directly to database in CDM

What is it not?

- No point-and-click interface
- Won’t construct T-C-O cohorts



CohortMethod package

Two layers:

- ‘Single study’
- ‘Multiple analyses’

Fully described in the vignettes included in the package



Single studies approach

```
cmData <- getDbCohortMethodData(...)
```

```
studyPop <- createStudyPopulation(cmData, ...)
```

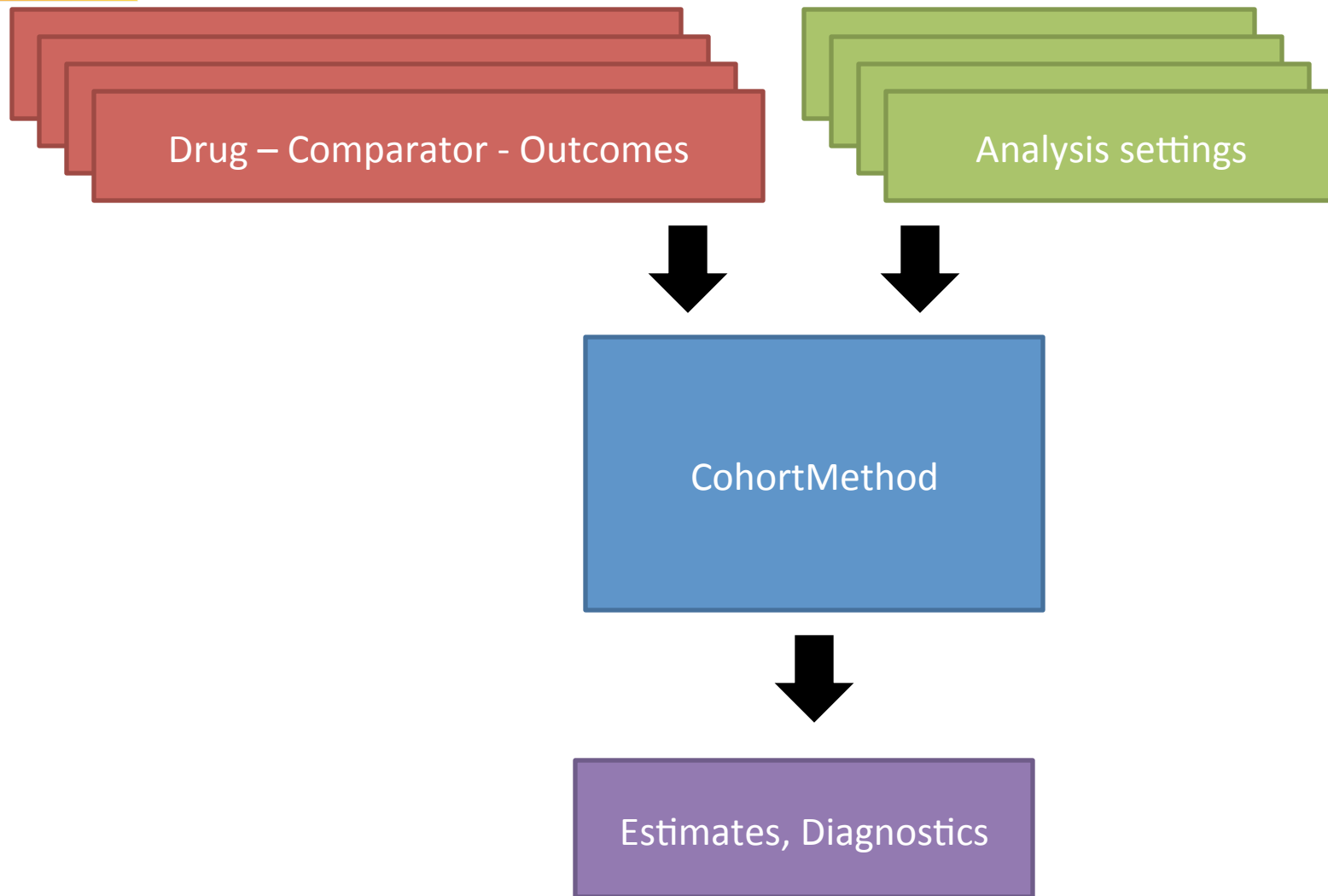
```
ps <- createPs(studyPop, cmData, ...)
```

```
matchedPop <- matchByPs(ps, ...)
```

```
model <- fitOutcomeModel(matchedPop, ...)
```



Running multiple analyses





Multiple analyses approach

```
cmdArgs <- createGetDbCohortMethodDataArgs(...)  
spArgs <- createCreateStudyPopulationArgs(...)  
psArgs <- createCreatePsArgs(...)  
matchArgs <- createMatchByPsArgs(...)  
modelArgs <- createFitOutcomeModelArgs(...)
```



```
analysis1 <- createCmAnalysis(analysisId = 1,  
                               description = "1-on-1 matching",  
                               cmdArgs,  
                               spArgs,  
                               psArgs,  
                               matchArgs,  
                               modelArgs,  
                               ...)
```



ATLAS editor creates analysis object



ATLAS

analysis1





Reproducing Graham et al. 2014
using the single study approach



Step 1: Getting the necessary data from the database

- Target and comparator cohorts of interest
 - Index date
 - End of cohort eligibility (e.g. end of exposure)
 - End of observation
- Outcome(s) of interest
 - Dates of occurrence
- Covariates

Take as-is from ATLAS



Covariates (Graham et al.)

Graham et al. used several covariates, including

- Demographics
- Medical history
 - Various conditions
 - CHADS
 - HAS-BLED
- Medication use

These were hand-picked for this study



Connecting to the database

```
connectionDetails <- createConnectionDetails(dbms = "pdw",  
                                              server = "JRDUSAPCTL01",  
                                              port = 17001)  
  
cdmDatabaseSchema <- "CDM_Truven_MDCR_V415.dbo"  
cdmVersion <- "5"
```



Loading the list of selected covariate concept IDs

```
includedConcepts <- read.csv("inst/csv/GrahamIncludedConcepts.csv")$Concept.ID
```



Covariate settings

```
covariateSettings <- createCovariateSettings(useCovariateDemographics = TRUE,  
                                             useCovariateDemographicsGender = TRUE,  
                                             useCovariateDemographicsRace = TRUE,  
                                             useCovariateDemographicsEthnicity = TRUE,  
                                             useCovariateDemographicsAge = TRUE,  
                                             useCovariateDemographicsYear = FALSE,  
                                             useCovariateDemographicsMonth = FALSE,  
                                             ...  
                                             useCovariateRiskScores = TRUE,  
                                             useCovariateRiskScoresCharlson = FALSE,  
                                             useCovariateRiskScoresDCSI = FALSE,  
                                             useCovariateRiskScoresCHADS2 = TRUE,  
                                             useCovariateRiskScoresCHADS2VAsC = FALSE,  
                                             excludedCovariateConceptIds = c(),  
                                             includedCovariateConceptIds = includedConcepts,  
                                             deleteCovariatesSmallCount = 100)
```

Including only the concept IDs
we just loaded





Loading all data

```
cohortMethodData <- getDbCohortMethodData(connectionDetails = connectionDetails,  
                                           cdmDatabaseSchema = cdmDatabaseSchema,  
                                           targetId = 2649,  
                                           comparatorId = 2650,  
                                           outcomeIds = 2651,  
                                           exposureTable = "cohort",  
                                           outcomeTable = "cohort",  
                                           cdmVersion = cdmVersion,  
                                           excludeDrugsFromCovariates = FALSE,  
                                           covariateSettings = covariateSettings)  
  
saveCohortMethodData(cohortMethodData, "s:/GrahamStudy/cohortMethodData")
```



Loading all data

```
cohortMethodData <- getDbCohortMethodData(connectionDetails = connectionDetails,  
                                           cdmDatabaseSchema = cdmDatabaseSchema,  
                                           targetId = 2649,  
                                           comparatorId = 2650,  
                                           outcomeIds = 2651,  
                                           exposureTable = "cohort",  
                                           outcomeTable = "cohort",  
                                           cdmVersion = cdmVersion,  
                                           excludeDrugsFromCovariates = FALSE,  
                                           covariateSettings = covariateSettings)  
  
saveCohortMethodData(cohortMethodData, "cohortMethodData")
```

Using the cohorts generated by ATLAS



Loading all data

```
cohortMethodData <- getDbCohortMethodData(connectionDetails = connectionDetails,  
                                           cdmDatabaseSchema = cdmDatabaseSchema,  
                                           targetId = 2649,  
                                           comparatorId = 2650,  
                                           outcomeIds = 2651,  
                                           exposureTable = "cohort",  
                                           outcomeTable = "cohort",  
                                           cdmVersion = cdmVersion,  
                                           excludeDrugsFromCovariates = FALSE,  
                                           covariateSettings = covariateSettings)
```

```
saveCohortMethodData(cohortMethodData, connectionDetails = connectionDetails,
```

The covariate settings we specified earlier



Inspecting the data

```
summary(cohortMethodData)
```

CohortMethodData object summary

Treatment concept ID: 2649

Comparator concept ID: 2650

Outcome concept ID(s): 2651

Treated persons: 19046

Comparator persons: 50918

Outcome counts:

	Event count	Person count
2651	8063	8063

Covariates:

Number of covariates: 827

Number of non-zero covariate values: 2428813



Step 2: Defining the study population

```
studyPop <- createStudyPopulation(cohortMethodData = cohortMethodData,  
                                outcomeld = 2651,  
                                removeSubjectsWithPriorOutcome = TRUE,  
                                minDaysAtRisk = 1,  
                                riskWindowStart = 1,  
                                addExposureDaysToStart = FALSE,  
                                riskWindowEnd = 0,  
                                addExposureDaysToEnd = TRUE)
```



Step 2: Defining the study population

```
studyPop <- createStudyPopulation(cohortMethodData = cohortMethodData,  
                                outcomeld = 2651,  
                                removeSubjectsWithPriorOutcome = TRUE,  
                                minDaysAtRisk = 1,  
                                riskWindowStart = 1,
```

Remove subjects with outcomes prior to
index date



Step 2: Defining the study population

```
studyPop <- createStudyPopulation(cohortMethodData = cohortMethodData,  
                                outcomeld = 2651,  
                                removeSubjectsWithPriorOutcome = TRUE,  
                                minDaysAtRisk = 1,  
                                riskWindowStart = 1,  
                                addExposureDaysToStart = FALSE,  
                                riskWindowEnd = 0,  
                                addExposureDaysToEnd = TRUE)
```

Define risk window: start on day after index date, end at end of cohort eligibility (end of exposure)

Graham: "Follow-up began on the day after the first qualifying anticoagulant prescription fill"



Step 3: Creating a propensity model

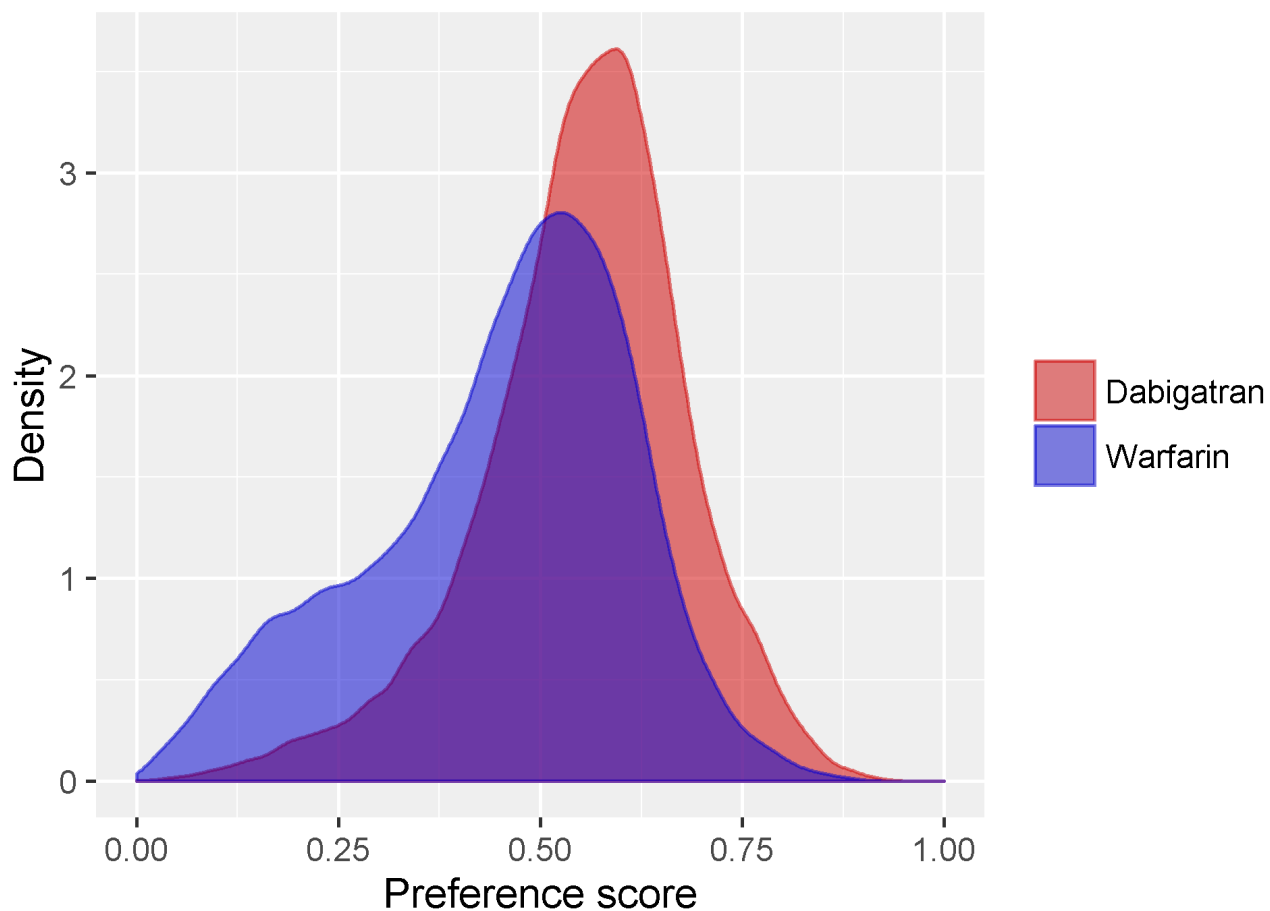
```
ps <- createPs(cohortMethodData = cohortMethodData,  
              population = studyPop)
```

```
saveRDS(ps, "s:/GrahamStudy/ps.rds")
```



Plot propensity score distribution

```
plotPs(ps , treatmentLabel = "Dabigatran", comparatorLabel = "Warfarin")
```





Step 4: Matching

```
matchedPop <- matchOnPs(ps,  
  caliper = 0.25,  
  caliperScale = "standardized",  
  maxRatio = 1)
```

Select up to 1 comparator per target subject
(1-on-1 matching)

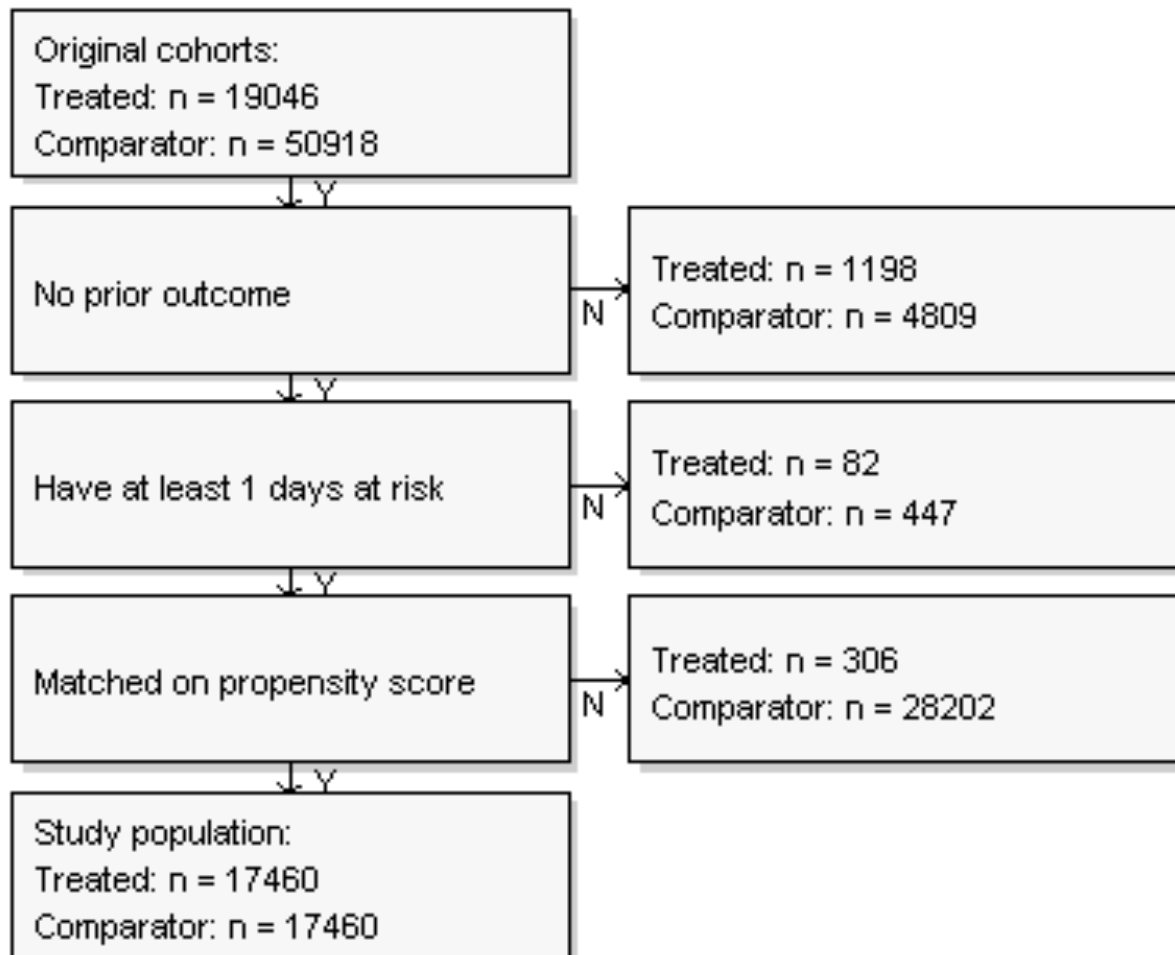
Graham: “Dabigatran users were propensity score matched to warfarin users in a 1:1 ratio with the use of a greedy matching algorithm.”

No caliper was mentioned, but it is probably a good idea to use one



Attrition

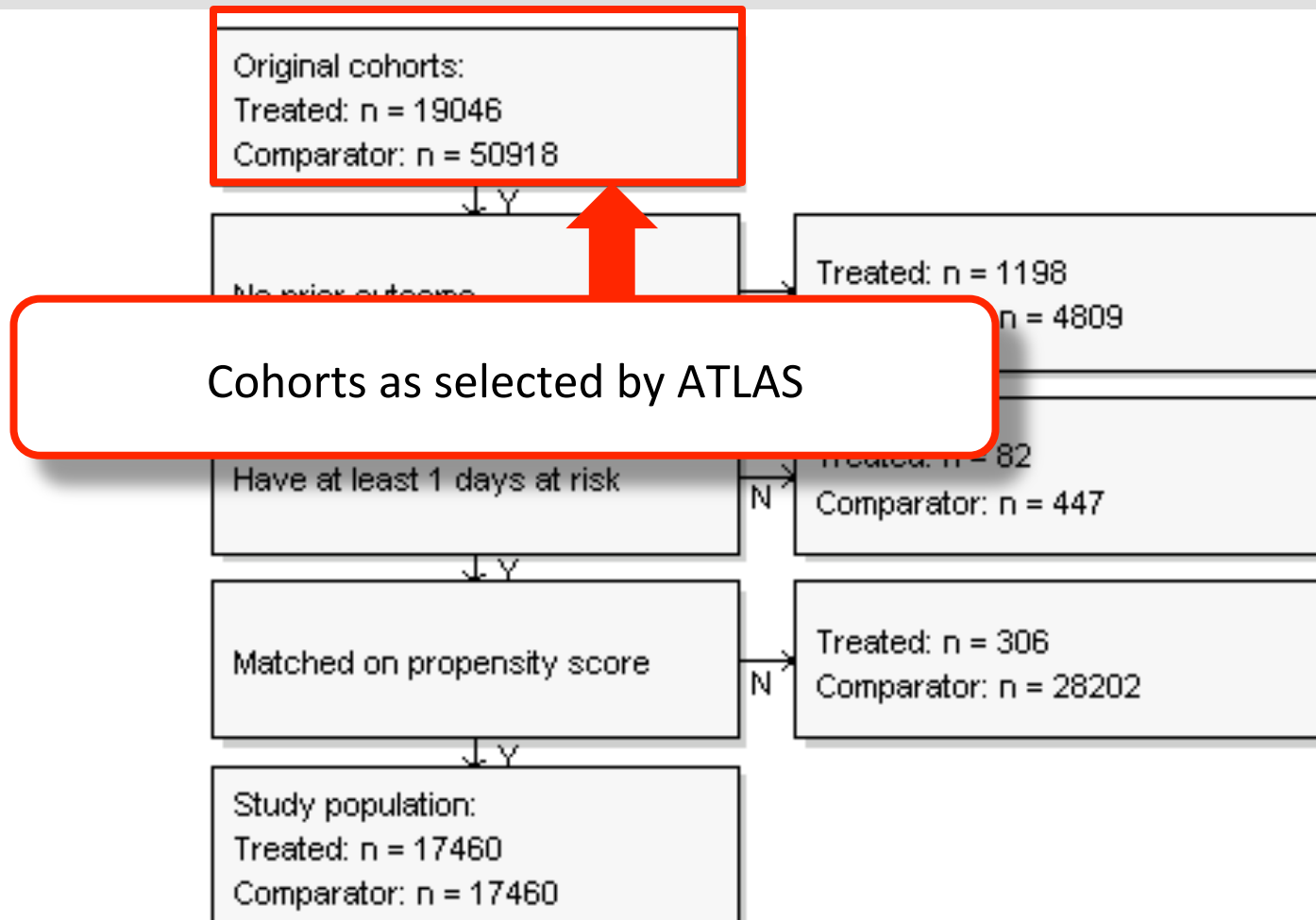
```
drawAttritionDiagram(matchedPop , treatmentLabel = "Dabigatran", comparatorLabel = "Warfarin")
```





Attrition

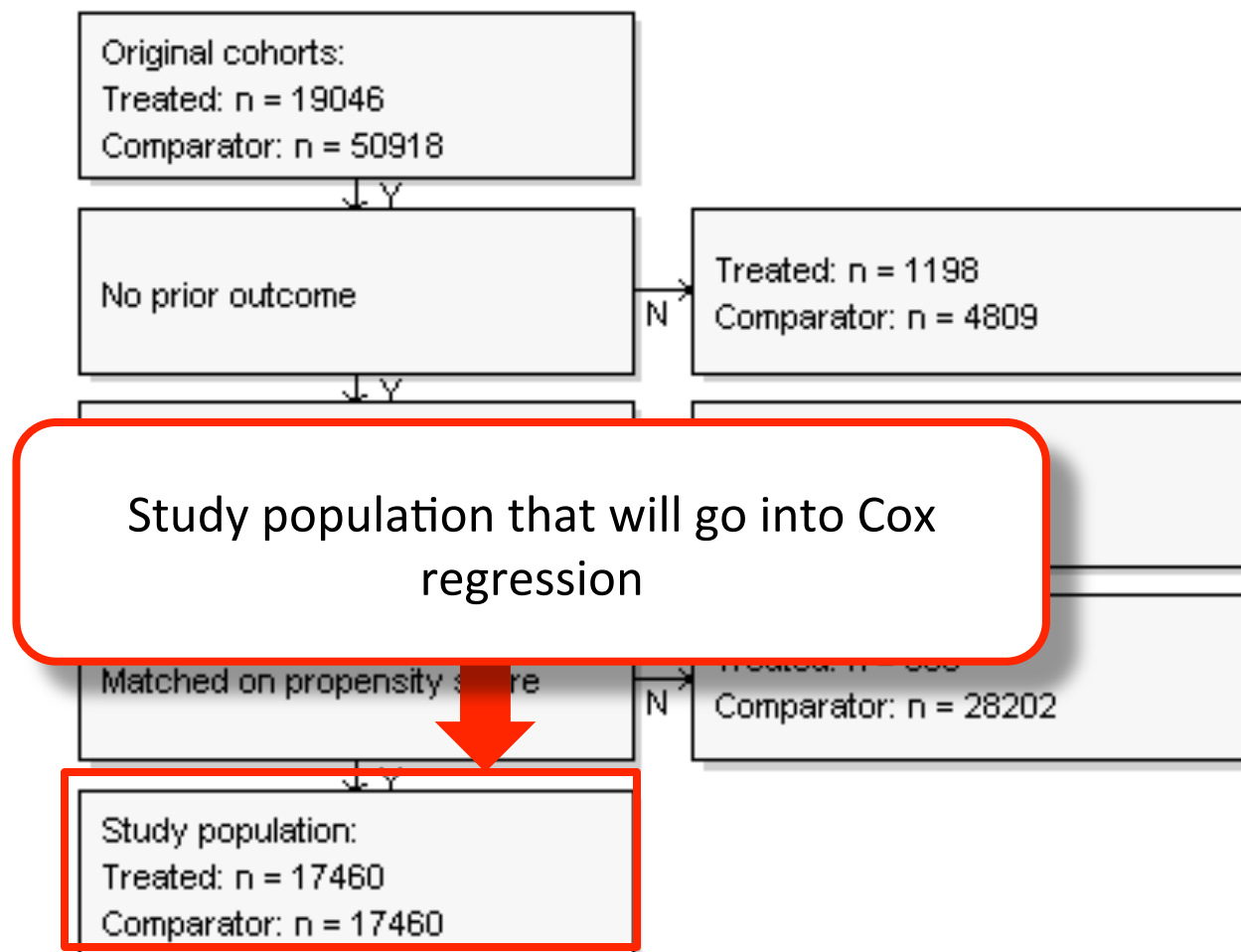
```
drawAttritionDiagram(matchedPop , treatmentLabel = "Dabigatran", comparatorLabel = "Warfarin")
```





Attrition

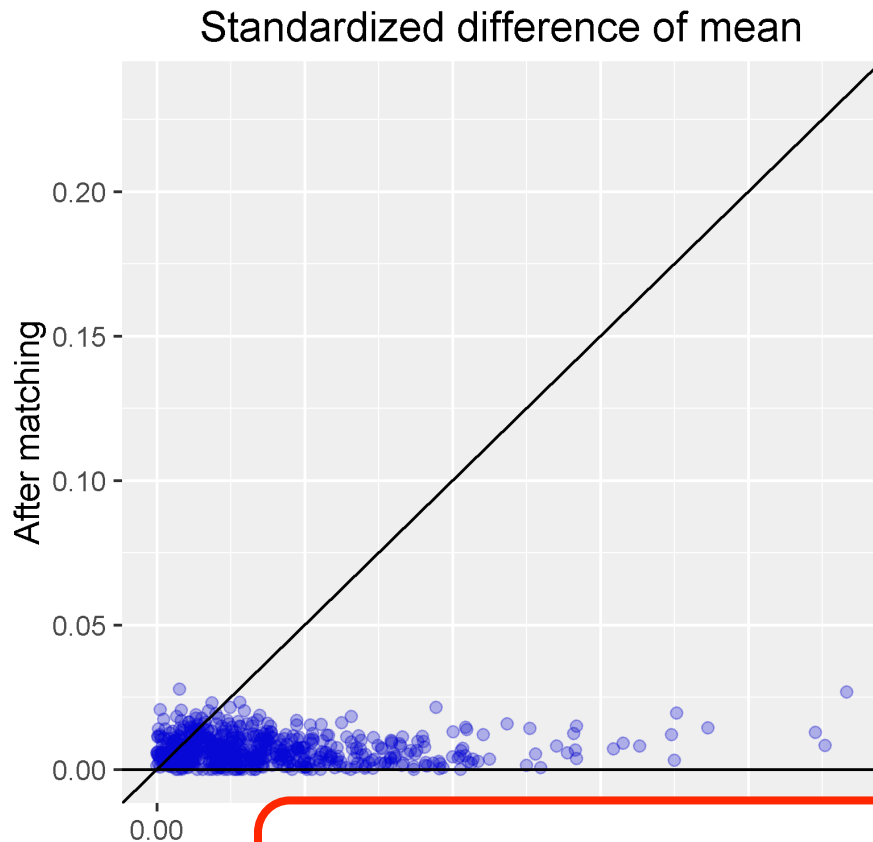
```
drawAttritionDiagram(matchedPop , treatmentLabel = "Dabigatran", comparatorLabel = "Warfarin")
```





Covariate balance

```
balance <- computeCovariateBalance(matchedPop, cohortMethodData)  
plotCovariateBalanceScatterPlot(balance)
```



Most covariates are binary:

$$\frac{\text{abs}(P_{\text{target group}} - P_{\text{comparator group}})}{\text{standard deviation}}$$

Graham: “A standardized mean difference of ≤ 0.1 indicates a negligible difference.”



Step 5: Fitting the outcome model

```
outcomeModel <- fitOutcomeModel(population = matchedPop,  
                                modelType = "cox",  
                                stratified = FALSE,  
                                useCovariates = FALSE)
```

Graham: “Cox proportional hazards regression was used to compare time to event in dabigatran compared with warfarin (reference) cohorts.

Ambiguous: Did they condition the model on the matched set (recommended)?



Inspect the outcome model

```
summary(outcomeModel)
```

Model type: cox
Stratified: FALSE
Use covariates: FALSE
Status: OK

	Estimate	lower .95	upper .95	logRr	seLogRr
treatment	0.89626	0.71863	1.11829	-0.10952	0.1128

Population counts

	treatedPersons	comparatorPersons	treatedExposures	comparatorExposures
Count	17460	17460	17460	17460

Outcome counts

	treatedPersons	comparatorPersons	treatedExposures	comparatorExposures
Count	164	155	164	155

Time at risk

	treatedDays	comparatorDays
Days	4912947	3954046



Inspect the outcome model

```
summary(outcomeModel)
```

Model type: cox

Stratified: FALSE

Use covariates: FALSE

Status: OK

	Estimate	lower .95	upper .95	logRr	seLogRr
treatment	0.89626	0.71863	1.11829	-0.10952	0.1128

Population counts

	treatedPersons	comparatorPersons	treatedExposures	comparatorExposures
Count	164	155	164	155

Point estimate and 95% confidence interval

	treatedDays	comparatorDays
Days	4912947	3954046

Time at risk

	treatedDays	comparatorDays
Days	4912947	3954046



Inspect the outcome model

```
summary(outcomeModel)
```

Model type: cox
Stratified: FALSE
Use covariates: FALSE
Status: OK

	Estimate	lower .95	upper .95	logRr	seLogRr
treatment	0.89626	0.71863	1.11		
Population counts					
	treatedPersons		com		es
Count	17460		1746		
Outcome counts					
	treatedPersons		comparatorPersons	treat	Exposures
Count	164		155	164	comparatorExposures
Time at risk					
	treatedDays		comparatorDays		
Days	4912947		3954046		

Target group (dabigatran) has more outcomes, but also more time at risk



Inspect the outcome model

```
summary(outcomeModel)
```

Model type: cox
Stratified: FALSE
Use covariates: FALSE
Status: OK

	Estimate	lower .95	upper .95
treatment	0.89626	0.71863	1.11829

Population counts

	treatedPersons	comparatorPersons
Count	17460	17460

Outcome counts

	treatedPersons	comparatorPersons	treatedExposures	comparatorExposures
Count	164	155	164	155

Time at risk

	treatedDays	comparatorDays
Days	4912947	3954046

Graham:

IR_{dabigatran} = 11.3

IR_{warfarin} = 13.9

HR_{adjusted} = 0.80 (0.67–0.96)

IR_{dabigatran} = 12.2

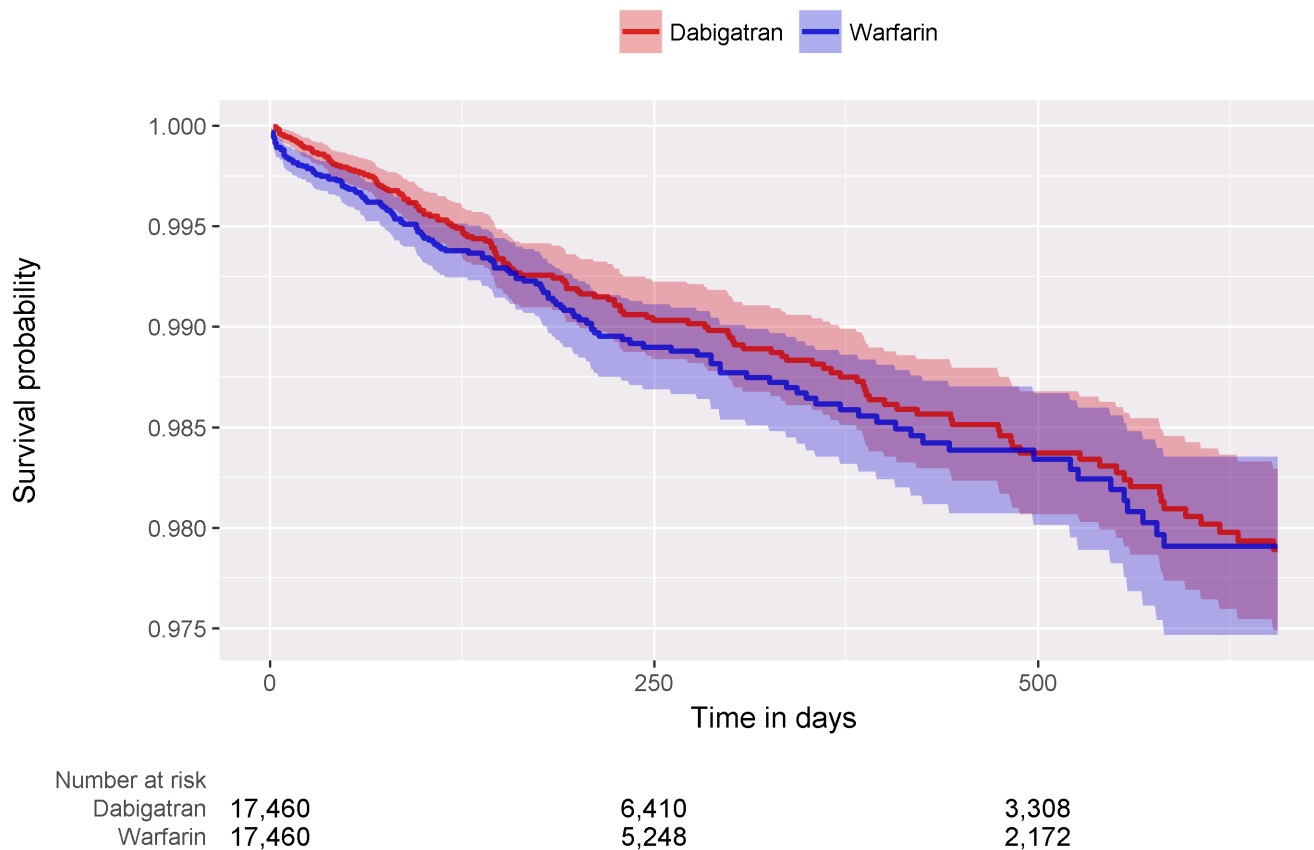
IR_{warfarin} = 14.3

HR_{adjusted} = 0.90 (0.72 – 1.12)



Kaplan Meier plot

```
plotKaplanMeier(matchedPop,  
  treatmentLabel = "Dabigatran",  
  comparatorLabel = "Warfarin")
```

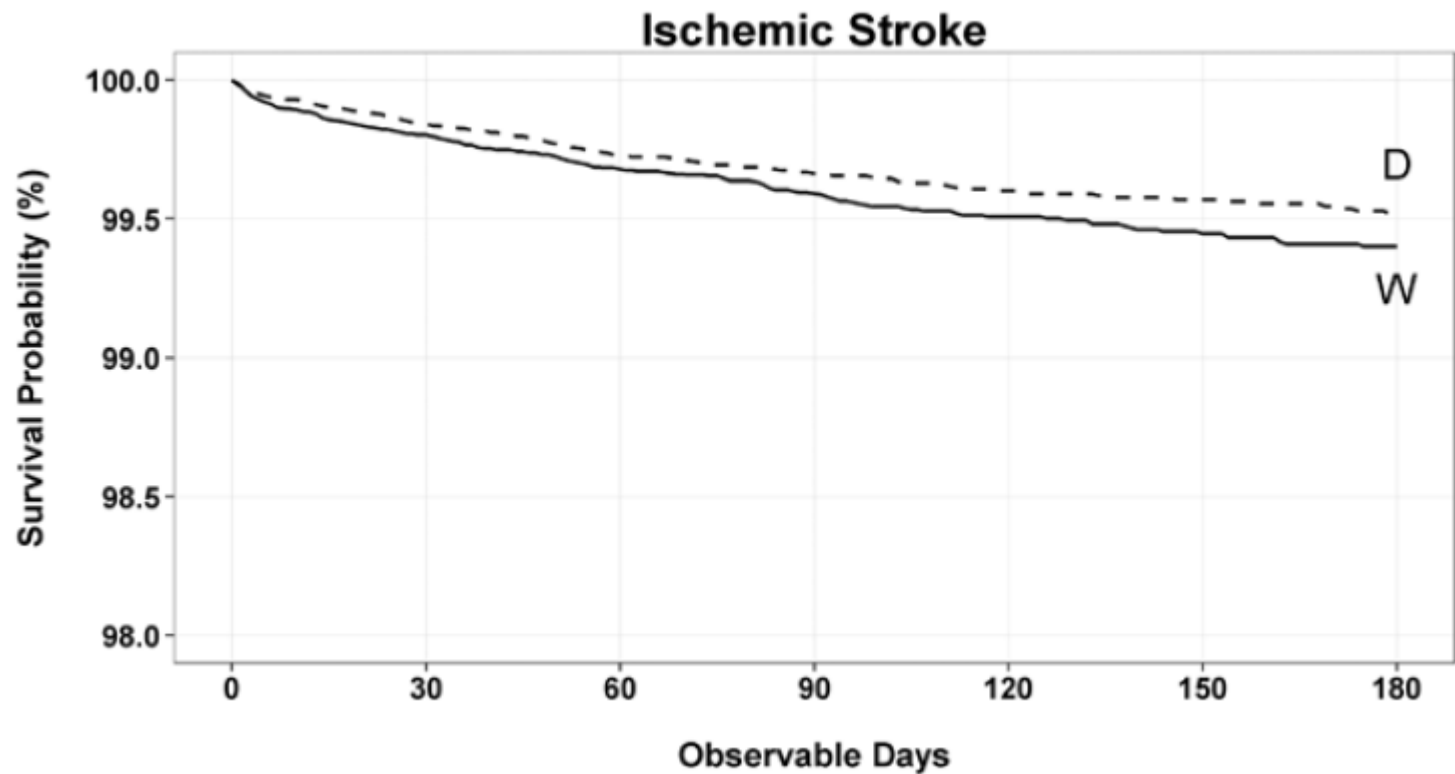




Kaplan Meier plot

```
plotKaplanMeier(matchedPop,
```

Graham:





Execution steps

1. Getting the necessary data from the database
 2. Defining the study population
 3. Creating a propensity model
 4. Matching
 5. Fitting the outcome model
- + generating various diagnostics



Review of the R study package generated by ATLAS



Anatomy of the study package

Execution code

Main execution code

Shiny app to view results

Cohorts

JSON definitions

SQL definitions

Negative controls

Negative control concepts

SQL template

Positive controls

Positive control settings

Analysis settings

Target-comparator-outcomes

Analysis definitions



Anatomy of the study package

Execution code

R

Main execution code

inst/shiny

Shiny app to view results

Cohorts

inst/cohorts

JSON definitions

inst/sql

SQL definitions

Negative controls

inst/settings

Negative control concepts

inst/sql

SQL template

Positive controls

inst/settings

Positive control settings

Analysis settings

inst/settings

Target-comparator-outcomes

inst/settings

Analysis definitions



How to install?

- Build and run in R-Studio
 - Open package
 - Own machine: unzip, double-click .Rproj file
 - R-Studio Server: upload zip, click .Rproj file
 - Install dependencies (see readme)
 - ‘Build’ → ‘Install and Restart’
- Build from GitHub
 - Unzip and put in GitHub repo
 - Install dependencies (see readme)
 - Install with `devtools::install_github`



Installing dependencies

How to run

1. In `R`, use the following code to install the dependencies:

```
install.packages("devtools")
library(devtools)
install_github("ohdsi/SqlRender", ref = "v1.5.2")
install_github("ohdsi/DatabaseConnector", ref = "v2.2.0")
install_github("ohdsi/OhdsiSharing", ref = "v0.1.3")
install_github("ohdsi/FeatureExtraction", ref = "v2.1.5")
install_github("ohdsi/CohortMethod", ref = "v3.0.1")
install_github("ohdsi/EmpiricalCalibration", ref = "v1.3.6")
install_github("ohdsi/MethodEvaluation", ref = "v0.3.1")
```

Source: readme.md



Running package

Source: readme.md

3. Once installed, you can execute the study by modifying and using the following code:

```
library(Graham)

# Optional: specify where the temporary files (used by the ff package) will be created:
options(fftempdir = "c:/FFtemp")

# Maximum number of cores to be used:
maxCores <- parallel::detectCores()

# Minimum cell count when exporting data:
minCellCount <- 5

# The folder where the study intermediate and result files will be written:
outputFolder <- "c:/Graham"

# Details for connecting to the server:
# See ?DatabaseConnector::createConnectionDetails for help
connectionDetails <- DatabaseConnector::createConnectionDetails(dbms = "postgresql",
  server = "some.server.com/ohdsi",
  user = "joe",
  password = "secret")

# The name of the database schema where the CDM data can be found:
cdmDatabaseSchema <- "cdm_synpuf"
```

Export data model

Study specification

analyses
cohort_method_analysis
- analysis_id
- description
- definition
covariate_analysis
- covariate_analysis_id
- covariate_analysis_name
exposures
exposure_of_interest
- exposure_id
- exposure_name
- definition
outcomes
outcome_of_interest
- outcome_id
- outcome_name
- definition
negative_control_outcome
- outcome_id
- outcome_name
positive_control_outcome
- outcome_id
- outcome_name
- exposure_id
- negative_control_id
- effect_size

metadata
database
- database_id
- database_name
- description
- is_meta_analysis
exposure_summary
- database_id
- exposure_id
- min_date
- max_date
comparison_summary
- database_id
- target_id
- comparator_id
- min_date
- max_date
attrition
- database_id
- exposure_id
- [target_id]
- [comparator_id]
- [outcome_id]
- [analysis_id]
- sequence_number
- description
- subjects*
cm_follow_up_dist
- database_id
- target_id
- comparator_id
- outcome_id
- analysis_id
- target_min_days
- target_p10_days
- target_p25_days
- target_median_days
- target_p75_days
- target_p90_days
- target_max_days
- comparator_min_days
- comparator_p10_days
- comparator_p25_days
- comparator_median_days
- comparator_p75_days
- comparator_p90_days
- comparator_max_days
covariate
- database_id
- covariate_id
- covariate_name
- covariate_analysis_id

underscore indicates primary key

[] indicates nullable

* indicates fields with a minimum value to avoid identifiability

Generated results

main results
cohort_method_result
- database_id
- target_id
- comparator_id
- outcome_id
- analysis_id
- rr
- ci_95_lb
- ci_95_ub
- p
- [i_2]
- log_rr
- se_log_rr
- target_subjects*
- comparator_subjects*
- target_days
- comparator_days
- target_outcomes*
- comparator_outcomes*
- calibrated_p
- calibrated_rr
- calibrated_ci_95_lb
- calibrated_ci_95_ub
- calibrated_p
- calibrated_rr
- calibrated_ci_95_lb
- calibrated_ci_95_ub
cm_interaction_result
- database_id
- target_id
- comparator_id
- outcome_id
- analysis_id
- interaction_covariate_id
- rrr
- ci_95_lb
- ci_95_ub
- p
- [i_2]
- log_rrr
- se_log_rrr
- target_subjects*
- comparator_subjects*
- target_days
- comparator_days
- target_outcomes*
- comparator_outcomes*
diagnostics
covariate_balance
- database_id
- target_id
- comparator_id
- [outcome_id]
- [analysis_id]
- [interaction_covariate_id]
- covariate_id
- target_mean_before*
- comparator_mean_before*
- std_diff_before
- target_mean_after*
- comparator_mean_after*
- std_diff_after
preference_score_dist
- database_id
- target_id
- comparator_id
- preference_score
kaplan_meier_dist
- database_id
- target_id
- comparator_id
- outcome_id
- analysis_id
- time
- [target_at_risk*]
- [comparator_at_risk*]
- target_survival
- target_survival_lb
- target_survival_ub
- comparator_survival
- comparator_survival_lb
- comparator_survival_ub
propensity_model
- database_id
- target_id
- comparator_id

Can be shared:

- Aggregated, so no patient-level data
- Minimum cell count enforced
- Saved as CSV, so easily reviewable
- Zipped for convenience

See vignette at <https://github.com/ohdsi/SkeletonComparativeEffectStudy> for details



Launching the Shiny app

5. To view the results, use the Shiny app:

```
prepareForEvidenceExplorer("Result<databaseId>.zip", "/shinyData")  
launchEvidenceExplorer("/shinyData", blind = TRUE)
```

Source: readme.md



Evidence Explorer





Concluding remarks

- CohortMethod package + R offer large flexibility
- 80% of studies are 'cookie-cutter' design, supported by ATLAS
- For remaining 20%, will need to modify code generated by ATLAS