Deconfounder with single outcome

To begin, install the following libraries from OHDSI github. Packages only need to be installed once.

```
devtools::install_github("ohdsi/SqlRender")
devtools::install_github("ohdsi/DatabaseConnector")
devtools::install_github("ohdsi/FeatureExtraction")
devtools::install_github("ohdsi/PatientLevelPrediction")
```

Connect to your database using the *DatabaseConnector* package. For details about the *createConnectionDetails* function, run ?createConnectionDetails or help(createConnectionDetails) in the R console.

```
connectionDetails = DatabaseConnector::createConnectionDetails(dbms = "sql server",
                                                server = "omop.dbmi.columbia.edu")
connection = DatabaseConnector::connect(connectionDetails)
```

Specify the following database schemas. The *targetCohortTable* is the name of the cohort table. Change the *targetCohortId* when create a new cohort.The *drugExposureTable* is the name of the table where drug exposure of the cohort will be stored. The *measurementTable* is the name of the table where both pre-exposure and post-exposure measurements of the cohort will be stored.

```
cdmDatabaseSchema = "ohdsi_cumc_deid_2020q2r2.dbo"
cohortDatabaseSchema = "ohdsi_cumc_deid_2020q2r2.results"
targetCohortTable = "MVDECONFOUNDER_COHORT"
drugExposureTable = "SAMPLE_COHORT_DRUG_EXPOSURE"
measurementTable = "SAMPLE_COHORT_MEASUREMENT"
targetCohortId = 1
```

The *conditionConceptIds* is the conditions of interest. The date of diagnosis of the condition is the cohort start date. The *measurementConceptId* is the outcome of interest. Currently outcome only supports lab measurement (continuous). For example, if the study is to estimate the treatment effect of drugs taken by a potassium disorder (both hypo- and hyperkalemia) cohort,

```
conditionConceptIds <- c(434610,437833) # Hypo and hyperkalemia
measurementConceptId <- c(3023103) # serum potassium
```

The *observationWindowBefore* *observationWindowAfter* are the time window (in days) to query for pre-treatment measurement and post-treatment measurement respectively. The *drugWindow* is the post-treatment time window (in days) to query for drug exposures from the DRUG_EXPOSURE table, and value 0 means only drugs prescribed on the same day as the day of diagnosis will be included. If drugWindow>0, then drugs prescribed post diagnosis will also be included.

```
observationWindowBefore <- 7
observationWindowAfter <- 30
drugWindow <- 0
```

To create the cohort and extract drug exposures and pre-treatment and post-treatment lab values. The output of *generateData* are two tables *measFilename* and *drugFilename* stored at *dataFolder*.

```
measFilename <- "meas.csv"
drugFilename <- "drug.csv"
dataFolder <- "path/to/datafolder"
MvDeconfounder::generateData(connection,
             cdmDatabaseSchema,
             oracleTempSchema = NULL,
             vocabularyDatabaseSchema = cdmDatabaseSchema,
             cohortDatabaseSchema,
             targetCohortTable,
             drugExposureTable,
             measurementTable,
             conditionConceptIds,
             measurementConceptId,
             observationWindowBefore,
             observationWindowAfter,
             drugWindow,
             createTargetCohortTable = T,
             createTargetCohort = T,
             extractFeature = T,
             targetCohortId=targetCohortId,
             dataFolder,
             drugFilename,
             measFilename)
```

The rest of the algorithm is implemented with python. First, specify the python to use using the *reticulate* package. For more

```
reticulate::use_condaenv("deconfounder_py3", required = TRUE)
```

First, preprocess the data for the deconfounder.

```
MvDeconfounder::preprocessingData(dataFolder, measFilename, drugFilename, drugWindow)
```

Specify the factor model to use, currently supporting "PMF" (poisson matrix factoriation) and "DEF" (two-layer deep exponential family).

```
factorModel <- 'DEF'
outputFolder <- "path/to/outputFolder"
```

Next, fit the deconfounder to estimate average treatment effect (ATE).

```
MvDeconfounder::fitDeconfounder(data_dir=dataFolder,
                save_dir=outputFolder,
                factor_model=factorModel,
                learning_rate=0.0001,
                max_steps=as.integer(100000),
                latent_dim=as.integer(1),
                layer_dim=c(as.integer(20), as.integer(4)),
                batch_size=as.integer(1024),
                num_samples=as.integer(1),
                holdout_portion=0.5,
                print_steps=as.integer(50),
```

```
                tolerance=as.integer(3),
                num_confounder_samples=as.integer(30),
                CV=as.integer(5),
                outcome_type='linear'
)
```

To visualize the estimated ATE, plot the mean and 95 CI as follows:

```
library(ggplot2)
resFolder <- "path/to/resultsFolder"
stats <- read.csv(file = file.path(resFolder, "treatment_effects_stats.csv"))

stats$drug_name <- factor(stats$drug_name, levels = stats$drug_name[order(-stats$mean)])
p2 <- ggplot(stats, aes(drug_name, mean)) + theme_gray(base_size=10)
p2 + geom_point(size=1) +
  geom_errorbar(aes(x = drug_name, ymin = ci95_lower, ymax = ci95_upper), width=0.2) +
  xlab("") +
  ylab("Estimated effect") +
  coord_flip()
```