PRS of Diabetes across UKBB

Muhammad Shoaib

18/10/2021

Summary of steps performed for defining diabetes cases (T1D and T2D) and controls in the UKBB data

1. A large phenotype file is subsetted based on the Patient IDs (first column) and ICD9 and ICD10 codes

2. The first 213 columns represent ICD10 codes with pattern 'f.41270*' and the remaining column represent ICD9 codes with pattern 'f.41271'* in the 'icdt' ukbb file

3. The first 214 columns were subsetted which comprised of Patient IDs and ICD10 codes

4. Here, we used a R package 'createPhenotypes'. This package requires ICD 9 and ICD 10 codes in the correct format

5. Since, we are dealing with ICD10 codes which are alphanumeric codes comprised of either three, four or five characters

6. UKBB ICD codes are in different format so we need to convert them in first the right format as required by the 'createPhenotype' package

7. Example ICD10 codes are in the UKBB are, (E16,E119, or possibly E23145). The right format of ICD10 codes become for analysis (E16, E11.9, E231.45)

8. The ICD 10 codes of interest are codes starting with 'E', as majority of them represent Diabetes
9. In the first step, we filtered all patient IDs with ICD10 codes starting with 'E'

10. In the second step, we place 'dot' after three characters in ICD10 codes required by R package for phecode conversion

11. After placing dots, there were majority of ICD10 codes which were not recognized by phecode_map_icd10 in the PheWAS package, so here I join the formatted ICD10 code column with the Phecode_map_IC10 column with correct codes using 'fuzzyjoin' package

12. I extracted the Patient IDs and correct ICD10 codes from the file and further use the 'count' function to count the IDs & code pair in file

13. In the final input file, 1st column = patient IDs, 2nd Column = vocabulary id which is 'ICD10', third column = ICD10 codes, 4th Column = count

14. Next we will apply, createPhenotypes function with minimum code count=1

15. In the results, we got a data frame with first column of Patient IDs and the rest of them are

Phecodes.Our Phecodes of interest are 250.1 (T1D) and 250.2 (T2D) which are cases

16. Exclusion phecodes are 249 and 250 and they need to be removed from Controls

17. Control IDs were obtained by excluding T1D, T2D and exclusion codes IDs from all UKBB ids which were almost half million

18. Next QC was performed based on ethnicities, kinship, sex discrepency and consent withdrawal

19. We consider european ancestry which comprised of white, British, all other white and Irish

R script for mapping ICD10 codes to Phecodes

```
salloc --time=1:0:0 --ntasks=1 --account=def-gsarah
cd projects/def-gsarah/shoaib/Diabetes-proj
module add r/4.0.5
R
library(data.table)
library(dplyr)
library(tidyverse)
library(PheWAS)
library(fuzzyjoin)
icdt <- fread("icd_codes.tab",header=T,sep="\t")
icdt <- as.data.frame(icdt)
icdt <- icdt[,c(1,(2:214))]   ## First 214 columns represents patients IDs and ICD10 columns ##

x1 <- icdt %>%
   filter(if_any(starts_with('f.41271'), ~ substr(., 1, 1) == '250')) %>%
   pivot_longer(cols = starts_with("f.41270"),
       values_to = 'DiagnosisCodes', names_to = NULL) %>%
   filter(substr(DiagnosisCodes, 1, 1) == "250")
x1 <- as.data.frame(x1)

ci_str_detect <- function(x, y) {
  str_detect(y, pattern = sub('(^[A-Z][0-9]{2})([0-9]{1,2})', '\\1\\.\\2', x, perl = TRUE))
}

x2 <- fuzzyjoin::fuzzy_left_join(x1, phecode_map_icd10, by = c("DiagnosisCodes" = "code"), match_fun = c

## In x2, only 2 columns are of interest, first (f.eid) and fourth one (code) ##

x3 <- x2[,c(1,4)]

colnames(x3)[2] <- 'DiagnosisCodes'

x4 <-  as.data.frame(count(x3, f.eid, DiagnosisCodes))

x5 <- data.frame(f.eid=x4$f.eid,vocabulary_id="ICD10",DiagnosisCodes=x4$DiagnosisCodes,count=x4$n)


x6 <- createPhenotypes(x5, min.code.count=1,
           add.phecode.exclusions=T, translate=T,
```

```
            full.population.ids=unique(x5[[1]]),
            aggregate.fun=PheWAS:::default_code_agg,
            vocabulary.map=PheWAS::phecode_map_icd10,
            rollup.map=PheWAS::phecode_rollup_map,
            exclusion.map=PheWAS::phecode_exclude)

## The results are TRUE/FALSE/NA ##

## For counting number of patients with T1D and T2D (T1D phecode=250.1, T2D phecode=250.2) ##

which(x6$'250.1'==TRUE)
which(x6$'250.2'==TRUE)

x7 <- as.data.frame(x6[,c(1,27)])
colnames(x7)[2] <- 'Codes'
x8 <- subset(x7, Codes)  ## T1D Patient IDs with TRUE values ##

x9 <- as.data.frame(x6[,c(1,33)])
colnames(x9)[2] <- 'Codes'
x10 <- subset(x9, Codes)  ## T2D Patient IDs with TRUE values ##

## All of T1D IDs are in x10 and they needs to be excluded ##

x11 <- anti_join(x10, x8, by=c("f.eid", "Codes"))

dim(x8)
= 4094 (T1D)

dim(x10)
= 30379 (T2D)

dim(x11)
= 27104 (UNIQUE T2D)

## Exclusion phecodes are 249 and 250, so we going to remove them from controls along with T1D and T2D

x12 <- as.data.frame(x6[,c(1,25)])  ## for exclusion 249 ##
colnames(x12)[2] <- 'Codes'
x12 <- subset(x12, Codes)

x13 <- as.data.frame(x6[,c(1,26)])  ## for exclusion 250 ##
colnames(x13)[2] <- 'Codes'
s13 <- subset(x13, Codes)

excl <- rbind(s12,s13)
dim(excl)
= 31205

## Here we will consider all UKBB ids and we will exclude exlusion codes, T1D, and T1D cases ##
c1 <- icdt[,c(1,2)]
chk <- anti_join(c1, x8, by=c("f.eid"))
chk1 <- anti_join(chk, x11, by=c("f.eid"))
chk2 <- anti_join(chk1, excl, by=c("f.eid"))
```

```
dim(chk2)  ##Controls##
=471290
```

QC steps

```
## QC based on ethnicities, and Kinship, data provided by Qiang"
testT2 <- semi_join(NT2Duniq, eth, by=c("f.eid"))
dim(testT2)
= 15284

testT1 <- semi_join(NT1D, eth, by=c("f.eid"))
dim(testT1)
= 2383

controls <- semi_join(cont3, eth, by=c("f.eid"))
dim(controls)
= 297717

## QC based on sex discrepency, data provided by Qiang"
T1sex <- anti_join(testT1, sex, by=c("f.eid"))
dim(T1sex)
= 2383

T2sex <- anti_join(testT2, sex, by=c("f.eid"))
dim(T2sex)
= 15284

Controlsex <- anti_join(controls, sex, by=c("f.eid"))
dim(Controlsex)
= 297717

## QC based on consent withdraw, data provided by Sarah, these are the final samples of Cases and Contr
T1consent <- anti_join(T1sex, withd, by=c("f.eid"))
dim(T1consent)
= 2383

T2consent <- anti_join(T2sex, withd, by=c("f.eid"))
dim(T2consent)
= 15284

Controlconsent <- anti_join(Controlsex, withd, by=c("f.eid"))
dim(Controlconsent)
= 297700
```