Practical in Polygenic Risk Scoring

Palle Duun Rohde, Izel Fourie Sørensen & Peter Sørensen

2022-06-30

Introduction

In this practical we will be perform polygenic risk scoring based on simulated data. The data consist of phenotypes for a disease, covariables, and genetic marker data. The practicals will be a mix of theoretical and practical exercises in R that are used for illustrating/applying the theory presented in the lecture.

- Practical 1: Use R for downloading data
- Practical 2: Explore phenotype data
- Practical 3: Prepare genotype data for polygenic risk scoring
- Practical 4: Polygenic risk scoring using clumping and thresholding
- Practical 5: Polygenic risk scoring using Bayesian linear regression models

The practical is based on the R package `qgg`.

This package provides an infrastructure for efficient processing of large-scale genetic and phenotypic data including core functions for:

- fitting linear mixed models
- constructing genetic relationship matrices
- estimating genetic parameters (heritability and correlation)
- performing genomic prediction and genetic risk profiling
- single or multi-marker association analyses

Installation of the R package qgg:

You can install qgg from CRAN with:

```
install.packages("qgg")
```

You can install the latest version of qgg from github with:

```
library(devtools)
devtools::install_github("psoerensen/qgg")
```

Practical 1: Downloading the data using R

load required packages

```
library(data.table)
```

Read files directly from website (github repository)

```
pheno <- fread(input = "https://github.com/psoerensen/qgdata/raw/main/simulated_human_data/human.pheno"
    data.table = FALSE)
```

```
covar <- fread(input = "https://github.com/psoerensen/qgdata/raw/main/simulated_human_data/human.covar"
    data.table = FALSE)
```

Create directory for downloading files

```
dir.create("C:\\Users\\au223366\\Dropbox\\Projects\\Summer_course")
```

Set working directory for the downloaded files

```
setwd("C:\\Users\\au223366\\Dropbox\\Projects\\Summer_course")
```

Download bedfile, bimfile, famfile (plink format genotype files) from website (github repository)

```
download.file(url = "https://github.com/psoerensen/qgdata/raw/main/simulated_human_data/human.bed",
    mode = "wb", destfile = "human.bed")

download.file(url = "https://github.com/psoerensen/qgdata/raw/main/simulated_human_data/human.bim",
    destfile = "human.bim")

download.file(url = "https://github.com/psoerensen/qgdata/raw/main/simulated_human_data/human.fam",
    destfile = "human.fam")
```

Note that mode="wb" for downloading the human.bed file. Otherwise the bed file will be corrupted and results wrong.

Download pheno and covar files from website (github repository)

```
download.file(url = "https://github.com/psoerensen/qgdata/raw/main/simulated_human_data/human.pheno",
    destfile = "human.pheno")
download.file(url = "https://github.com/psoerensen/qgdata/raw/main/simulated_human_data/human.covar",
    destfile = "human.covar")
```

Practical 2: Preparing the phenotype data using R

```
library(data.table)
```

Prepare phenotype and covariables

```
pheno <- fread(input = "C:\\Users\\au223366\\Dropbox\\Projects\\Summer_course\\human.pheno",
    data.table = FALSE)
```

```
covar <- fread(input = "C:\\Users\\au223366\\Dropbox\\Projects\\Summer_course\\human.covar",
    data.table = FALSE)
```

One of the first thing to do is to explore the data used in the analysis. The goal is to understand the variables, how many records the data set contains, how many missing values, what is the variable structure, what are the variable relationships and more. Several commands/functions will be used. To read more about a specific function (e.g., str) write ?str.

Question 1: How many observations and which variables do we have in the data set? To get a fast overview of the data set you are working with you can use the str or head functions:

```
str(pheno)
```

```
## 'data.frame':    5000 obs. of  3 variables:
##  $ V1: chr  "IND1" "IND2" "IND3" "IND4" ...
##  $ V2: chr  "IND1" "IND2" "IND3" "IND4" ...
##  $ V3: int  0 0 1 0 0 1 0 0 1 1 ...
```

```
str(covar)
```

```
## 'data.frame':    5000 obs. of  14 variables:
##  $ V1 : chr  "IND1000" "IND1001" "IND1002" "IND1003" ...
##  $ V2 : chr  "IND1000" "IND1001" "IND1002" "IND1003" ...
##  $ V3 : int  1 0 0 0 0 1 1 1 1 0 ...
##  $ V4 : num  61.2 62.4 65.6 55.4 64.1 ...
##  $ V5 : num  0.00488 -0.0063 -0.00522 0.02028 -0.00411 ...
##  $ V6 : num  -0.002097 0.000148 -0.000606 0.006778 0.004349 ...
##  $ V7 : num  0.01091 0.00469 0.01709 -0.00306 -0.00839 ...
##  $ V8 : num  0.0056 0.01611 -0.00442 -0.01284 0.01022 ...
##  $ V9 : num  -0.008546 -0.00262 0.034218 -0.003078 -0.000912 ...
##  $ V10: num  0.019736 0.000308 -0.005157 0.012835 0.004788 ...
##  $ V11: num  -0.00408 0.00961 -0.00289 -0.01387 0.00977 ...
##  $ V12: num  0.01435 -0.01479 -0.00901 0.01499 0.00899 ...
##  $ V13: num  -0.00152 -0.00708 0.00706 0.00474 -0.00381 ...
##  $ V14: num  0.003998 0.037763 -0.000359 -0.022958 -0.010249 ...
```

```
head(pheno)
```

```
##     V1   V2 V3
## 1 IND1 IND1  0
```

```
## 2 IND2 IND2  0
## 3 IND3 IND3  1
## 4 IND4 IND4  0
## 5 IND5 IND5  0
## 6 IND6 IND6  1
```

```
head(covar)
```

```
##        V1      V2 V3       V4        V5        V6        V7        V8        V9
## 1 IND1000 IND1000  1 61.24445  0.004881 -0.002097  0.010908  0.005597 -0.008546
## 2 IND1001 IND1001  0 62.42271 -0.006301  0.000148  0.004692  0.016107 -0.002620
## 3 IND1002 IND1002  0 65.64274 -0.005215 -0.000606  0.017091 -0.004416  0.034218
## 4 IND1003 IND1003  0 55.40703  0.020284  0.006778 -0.003061 -0.012837 -0.003078
## 5 IND1004 IND1004  0 64.10666 -0.004105  0.004349 -0.008388  0.010221 -0.000912
## 6 IND1005 IND1005  1 65.77823 -0.010636 -0.021853 -0.017450  0.010058  0.000502
##         V10       V11       V12       V13       V14
## 1  0.019736 -0.004078  0.014350 -0.001518  0.003998
## 2  0.000308  0.009612 -0.014793 -0.007077  0.037763
## 3 -0.005157 -0.002894 -0.009013  0.007064 -0.000359
## 4  0.012835 -0.013872  0.014987  0.004744 -0.022958
## 5  0.004788  0.009773  0.008988 -0.003814 -0.010249
## 6  0.003807 -0.003007  0.005646 -0.007199 -0.009874
```

Question 2: How is the phenotype distributed? Use the histogram and boxplot functions to visualize the distribution the trait: Define the response variable

```
y <- pheno[, 3]
names(y) <- pheno[, 1]
```

The exploratory data analysis is the process of analyzing and visualizing the data to get a better understanding of the data. It is not a formal statistical test. Which factors should we include in the statistical model?

To best answer these question we can fit a logistic regression model that include these factors in the model.

This can be done using the glm function:

Fit logistic regression model

```
fit <- glm(y ~ V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V11 +
    V12 + V13 + V14, data = covar, family = binomial(link = "logit"))
summary(fit)
```

```
##
## Call:
## glm(formula = y ~ V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V11 +
##     V12 + V13 + V14, family = binomial(link = "logit"), data = covar)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9283  -0.7765  -0.7439   1.5279   1.8371
##
## Coefficients:
```

```
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.181957   0.214978  -5.498 3.84e-08 ***
## V3          -0.011066   0.065316  -0.169   0.8655
## V4           0.001740   0.003818   0.456   0.6487
## V5           1.309941   2.306100   0.568   0.5700
## V6          -0.280830   2.307468  -0.122   0.9031
## V7           2.354268   2.306546   1.021   0.3074
## V8          -1.349941   2.306494  -0.585   0.5584
## V9           5.520940   2.311308   2.389   0.0169 *
## V10         -2.724333   2.307482  -1.181   0.2377
## V11         -1.929366   2.307348  -0.836   0.4031
## V12          0.578919   2.308375   0.251   0.8020
## V13          4.382604   2.307287   1.899   0.0575 .
## V14         -0.504532   2.306778  -0.219   0.8269
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5645.2  on 4999  degrees of freedom
## Residual deviance: 5631.6  on 4987  degrees of freedom
## AIC: 5657.6
##
## Number of Fisher Scoring iterations: 4
```

Create design matrix for the explanatory variables

```
X <- model.matrix(~V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V11 +
    V12 + V13 + V14, data = covar)
rownames(X) <- covar$V1
X <- X[names(y), ]
sum(names(y) %in% rownames(X))
```

```
## [1] 5000
```

Define training and validation samples

```
train <- sample(names(y), 4000)
valid <- names(y)[!names(y) %in% train]
```