Bioinformatics - Homework 3 - Classic style
(Prof. K. Van Steen)

Contact person : Raphaël Liégeois (R.Liegeois@ulg.ac.be)

November 13, 2012

Instructions : *in this assignment you will test your understanding of basic GWAS concepts and GenABEL functions. The materials needed for the homework (two data files and an R function to be downloaded in your working directory) are available on K. Bessonov's homepage. You are asked to :*

- *Provide a single final report with the (concise) answers to the following questions*
- *Provide (in that report) any additional code you used in order to answer the questions*
- *Return your report before December 4, 2012*

---

1 Part 1 : Preprocessing the data

Preprocessing bioinformatics data is a mandatory step if one wants to perform a relevant analysis and highlight interesting relationships in the data. In this homework we will first try to get familiar with some GWA techniques and apply them on real data. We will then focus on the preprocessing steps at the sample level and at the marker level and we will finish with a brief association analysis using the cleaned data. To do so we will use the GenABEL R package for which a full tutorial is available on the website `http://mga.bionet.nsc.ru/~yurii/ABEL/GenABEL/`. The data are gathered in two distinct files corresponding to phenotypic (wgDat.phe) and genotypic (wgDat.raw) information and can be downloaded at `http://www.montefiore.ulg.ac.be/~kbessonov/courses.html`

After loading the GenABEL library you can load the data by running the following commands :

```
library(GenABEL)
pheFile <- "wgDat.phe"
rawFile <- "wgDat.raw"
wgDat <- load.gwaa.data(pheFile, rawFile)
```

If everything is loaded correctly you should get the following messages :

```
ids loaded...
marker names loaded...
chromosome data loaded...
```

```
map data loaded...
allele coding data loaded...
strand data loaded...
genotype data loaded...
snp.data object created...
assignment of gwaa.data object FORCED; X-errors were not checked!
```

Before going through the preprocessing steps it is important to get familiar with the data we are dealing with. Using some basic GenABEL commands such as `str()` or `perid.summary()` you should be able to get some information about both genotypic and phenotypic data.

Question 1 :

- *How many patients are represented in this study ?*
- *Here controls are coded with `affection = 1` and cases with `affection = 2`. How many controls are there in the sample ?*
- *What is the sex of the patient with id `"NA18945"` ?*
- *On which chromosome is the SNP `"rs10873910"` located ?*
- *Are there missing samples for this SNP ?*
- *What is the minor allele for that SNP ?*
- *Is the patient with id `"NA18529"` homozygous or heterozygous for that SNP ?*

Now that you are familiar with basic GenABEL functions we will go through the quality control steps. We will use the Travemünde criteria both at the sample and at the SNP (marker) level.

1.1 At the sample level

First let us focus on the two criteria about Call fraction (proportion of correctly identified genotypes) and Heterozigosity.

Question 2 :

- *What are the constraints on Call fraction and heterozigosity according to Travemünde criteria ?*
- *Why is the heterozygosity criterion important ?*
- *Fill in the following lines (in the three places marked "TO FILL") in order to eliminate undesirable samples according to the criteria determined at the two previous points. Determine how many samples were removed according to those criteria.*

```
idSummar <- perid.summary(wgDat)
head(idSummar)
hetMean <- mean(idSummar$Het)
hetSd <- sd(idSummar$Het)
hetThreshUpp <- TO FILL
hetThreshLow <- TO FILL
removeIdx <- with(idSummar, which(TO FILL))
idSummar$keep <- TRUE
idSummar$keep[removeIdx] <- FALSE
```

```
keepIDs <- row.names(idSummar[idSummar$keep, ])
wgDatIdClean <- wgDat[keepIDs, ]
```

Let us now consider the condition about ethnic origin. The point here is to check for so called "population stratification", i.e. to assess that there are no groups of different ethnic origins that could bias the data structure in our further analysis. If it is the case it is still possible to work with the data but you will have to select homogeneous subsets of individuals.

To do so we will use multidimensional scaling (MDS), a variant of principal component analysis (PCA), based on the pairwise similarity of the samples :

```
pwSim <- ibs(wgDatIdClean)
pwDist <- as.dist(0.5 - pwSim)
mdsDat <- cmdscale(pwDist)
plot(mdsDat[, 1], mdsDat[, 2], xlab = "Component 1",
       ylab = "Component 2", pch = 19)
```

Question 3 :

- *Explain the second command line computing pwDist. Why do we introduce the 0.5 term ?*
- *Is there population stratification in the data that should be taken into account ?*
- *Why do we use only two components in MDS ? Would it be useful to consider more components by using classical PCA for example ?*

1.2 At the SNP level

The second part of the preprocessing consists of removing the SNPs that does not fulfill the Travemünde criteria.

Question 4 :

- *What are the constraints on MAF, MiF*[1] *by group and HWE according to Travemünde criteria ?*
- *Explain those three criteria in your own words. What are the negative effects that could occur if those criteria are not met ?*
- *Fill in the following lines (in the three places marked "TO FILL") in order to eliminate undesirable SNPs according to the criteria defined at the first point*[2]*.*

```
casesIDs <- subset(wgDatIdClean@phdata, TO FILL , id, drop = TRUE)
controlsIDs <- subset(wgDatIdClean@phdata, TO FILL , id, drop = TRUE)
sumMaf <- summary(wgDat)$Q.2
maf <- data.frame(maf=sumMaf)
sumControls <- summary(wgDatIdClean[controlsIDs,
       ])[, c("Pexact", "CallRate")]
names(sumControls) <- c("pHWE", "cfControls")
```

1. MiF stands for Missing Frequency, i.e. the proportion of missing genotypes
2. Note that the departure from HWE is only computed in controls

```
sumCases <- summary(wgDatIdClean[casesIDs, ])[, "CallRate",
       drop = FALSE]
names(sumCases) <- "cfCases"
snpSummar <- do.call(cbind, list(maf, sumControls,
       sumCases))
snpRemoveIdx <- with(snpSummar, which(TO FILL))
snpSummar$keep <- TRUE
snpSummar$keep[snpRemoveIdx] <- FALSE
keepSNPs <- row.names(snpSummar[snpSummar$keep, ])
wgDatClean <- wgDatIdClean[, keepSNPs]
```

Then, the code in `plot-VennDiagram.R` allows to represent in a nice way the number of SNPs rejected and according to which criterion(a) :

```
source("plot-VennDiagram.R")
mafSNPs <- row.names(subset(snpSummar, maf < 0.01))
hweSNPs <- row.names(subset(snpSummar, pHWE < 1e-04))
crSNPs <- row.names(subset(snpSummar, cfCases < 0.98 |
       cfControls < 0.98))
qcVenn(mafSNPs, hweSNPs, crSNPs, labels = c("MAF",
       "HWE", "CF"), numberSnps = nrow(snpSummar))
```

Question 4b :

- *How many SNPs were eliminated according to the different criteria ?*
- *How many SNPs were excluded according to both the MAF and CF criteria ?*
- *Are the three criteria balanced for our dataset ? Should we adapt them to our data ?*

2 Part 2 : Introduction to association analysis

Now that the data is cleaned we can perform an association analysis. To do so, we first need to binarize the affection criterion :

```
wgDatClean@phdata$aff.01 <- wgDatClean@phdata$affection - 1
with(wgDatClean@phdata, table(affection, aff.01))
```

We then test the hypothesis of a link between the trait and every remaining SNP using a linear regression model :

```
assocRes <- mlreg(aff.01 ~ 1, data = wgDatClean,
       gtmode = "additive", trait.type = "binomial")
plot(assocRes, main = "", ystart = 2)
```

The graph you obtained is called a Manhattan plot for obvious reasons and represents on the y-axis the p-value related to the hypotheses that were tested.

Question 5 :

- *If we consider an association being meaningful when* $-log(P-value) > 4.5$*, are there SNPs that can be said as meaningfully related to the trait according to that criterion ?*
- *If yes, where are they located, what are their names, are they associated to a gene ?*