Course: GBIO0009 2015-16

Bioinformatics - Homework 1 – Q&A style

Instructions: in this assignment you will test your understanding of basic GWAS concepts and GenABEL functions. The materials needed for the homework (two datasets and R function) are all available on the course homepage. You are asked to:

*Provide a **single report per group** with the (concise) answers to the following questions (include Introduction, Discussion and Results and Conclusion sections).*
Provide any additional code you used (if any) in Appendix or a separate file

---

Part 1: Preprocessing the genotype data

Preprocessing bioinformatics-related 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 1st try to get familiar with some GWA techniques and apply them on a real data. We will then focus on the preprocessing steps (i.e. quality control) 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 at the http://mga.bionet.nsc.ru/~yurii/ABEL/GenABEL/. The data are split in two distinct files corresponding to phenotypic (**wgDat.phe**) and genotypic (**wgDat.raw**) information and can be downloaded at from the course website.

After loading the GenABEL library and downloading data, you should 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 further data 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 (tip: check the call rate value)?
- What is the minor allele for *rs10873910* bi-allelic SNP (tip: convert genotypes to allele counts)?
- 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 Travemunde criteria both at the sample and at the SNP (marker) level.

1.1 Sample (phenotypic) level

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

Question 2 :

- What are the constraints (i.e. thresholds) on the “call fraction” and “heterozygosity” according to the Travemünde criteria?
- Why is the heterozygosity criterion important in the genetics context? What are the main factors that could impact heterozygosity of a given population? Discuss in context of Hardy-Weinberg equilibrium (HWE)
- 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 (i.e. call fraction and heterozygosity). Determine how many samples were removed according to those criteria / filters.
    - Tip 1: in the 1st two “to fill” parts calculate upper and lower bound of the heterozygosity threshold
    - Tip 2: in the 3rd “to fill” insert all filtering criteria to select desired samples passing QC filters

```
idSummar <- perid.summary(wgDat);
hetMean <- mean(idSummar$Het);
hetSd <- sd(idSummar$Het;
hetThreshUpp <- TO FILL;
hetThreshLow <- TO FILL;

removeIdx <- with(idSummar, which(hetThreshLow>idSummar$Het,
hetThreshUpp<idSummar$Het, 0.97>idSummar$CallPP) );
idSummar$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 represented by kinship matrix:

```
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? Explain numbers in meaning of the pairwise similarity matrix *pwDist.*
- Is there population stratification in the data that should be taken into account? Tip: look at cluster homogeneity
- Why do we use only two components in MDS? Would it be useful to consider more components by using classical PCA, for example? Provide supporting arguments in each case

1.2 Gentotypes (SNP) level

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

Question 4 :

- What are the constraints/thresholds on MAF, MiF by group and HWE according to Travemünde criteria?
    - Note: MiF stands for Missing Frequency, i.e. the proportion of missing genotypes
- Explain those three criteria in your own words. What are the negative effects that could occur if those criteria are not met? What will be impact on the final results interpretation?
- 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 1st point.
    - Tip 1: use data frame column that allows to select cases and controls
    - Tip 2: in the 3rd to fill insert all filtering criteria selecting both SNPs to remove in cases and controls. There is a total of 3 criteria

```
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");

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, copy and paste the qcVenn function code in the Appendix to represent in a nice way the number of SNPs rejected and according to the criterion below:

```
#define qcVenn() here
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? Provide statistics per each criteria individually.
- How many SNPs were excluded according to both the MAF and CF criteria?
- Are the three criteria balanced/optimally selected for our dataset? Should we adapt them to our data? Tip: look at the Venn plot intersection regions

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 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 and represents on the y-axis the *p*-value related to the hypotheses that were tested involving potential association between genotype and disease status (i.e. trait)

Question 5:

- If we consider an association being meaningful (i.e. statistically significant) when log(p-value) > 4.5, are there SNPs that can be said as meaningfully related to the trait according to the chosen criterion? If any, where are they located, what are their names (SNP ids), are they associated to any gene? Provide the resulting Manhattan plot.

Appendix – the qcVenn() function definition

```
## Arne Schillert 07.07.2008
## new written Venn diagram
## inspired by library limma's venn.diagram

qcVenn <- function(A, B, C, labels, textSize = 1,
                   labelSize = 1.2, numberSnps)
{
  p1 <- length(setdiff(A, union(B, C)))
  p2 <- length(setdiff(intersect(A, B), C))
  p3 <- length(setdiff(B, union(A, C)))
  p4 <- length(setdiff(intersect(A, C), B))
  p5 <- length(intersect(intersect(A, B), C))
  p6 <- length(setdiff(intersect(B, C), A))
  p7 <- length(setdiff(C, union(A, B)))
  counts <- c(p1, p2, p3, p4, p5, p6, p7)

  ## circles:
  xCen <- c(-1, 1, 0)
  yCen <- c(1/sqrt(3), 1/sqrt(3), -2/sqrt(3))
  r <- 1.5
  theta <- 2 * pi * (1:720)/720
  par(mai = c(0, 0, 0, 0))
  plot(0, type = "n", xlim = c(-3.2, 3.2),
       ylim = c(-3.2,3.2), axes = FALSE,
       xlab = "", ylab = "")
  for(i in seq(along = xCen))
      {
        lines(xCen[i] + r * cos(theta),
              yCen[i] + r * sin(theta), lwd = 1.2)
      }
  xD <- 2
  yD <- sqrt(3)
  xBig <- 0.8
  xSmall <- 0.4
  yBig <- 0.7
  ySmall <- 0.4
  xPos <- c(xCen[1] - xSmall, 0, xCen[2] + xSmall,
            -xBig, 0, xBig, 0)
  yPos <- c(yCen[1] + ySmall, yCen[1] + ySmall, yCen[1] + ySmall,
            yCen[3] + yBig, yCen[2] - sqrt(5)/4,yCen[3] + yBig,
            yCen[3] -ySmall)
  text(xPos, yPos, counts, cex = textSize)
  text(c(-1.1, 1.1, 0), c(2.45, 2.45, -3.1),
       labels, cex = labelSize)
  nOut <- length(unique(c(A, B, C)))
  text(-1.7, 3.1 , paste("total:" ,numberSnps),
       cex = 1.2)
  text(2.1, 3.1 , paste("total excluded:" ,nOut),
       cex = 1.2)


  ## adding number of SNPs which passed each criterion
```

```
  ns <- c(length(A), length(B), length(C))


  for(i in 1:3){
    text(1.5, (-2.1 - 0.25 * i),
         sprintf("%s: ", labels[i]),
         cex = 1.2, adj = 0, font = 7)
    text(2.95, (-2.1 - 0.25 * i), ns[i],
         cex = 1.2, adj = 1)
  }

}
```