**Lab 7**. Statistical methods for finding associations between Single Nucleotide Polymorphisms (SNPs) and phenotypes (case/control) in genome-wide association study (gwas) data. We will use the chi-square test and logistic regression to test for association.

**0.** Reading PLINK files. Use the code below with the snpStats library to read the toy PLINK data in the files extra.ped and extra.map.

**1.** How many subjects and predictors are there (use dim)?

[in] dim(genotypes)

[out] 89 17

so 89 subjects with 17 predictors

**2.** How many phenotype-0 samples are heterozygous for the SNP below? How many phenotype-1 samples are heterozygous (see the output of the table)?

[in] table(phenotype,genotypes.df$rs630969,dnn=c("phenotype","genotype"))

[out]

phenotype A/A A/B B/B

0 11 17 13

1 10 21 17

so for phenotype-0 there are 17

library(snpStats) #BiocManager::install("snpStats")

ex.data <- read.pedfile(file="extra.ped", snps="extra.map")

ex.data$fam

phenotype <- ex.data$fam$affected-1 # change pheno from 1/2 to 0/1

genotypes <- ex.data$genotypes # encoded as AA/AB/BB

snp.ids <- as.character(ex.data$map$snp.names)

colnames(genotypes.df) <- snp.ids

genotypes.df <- data.frame(as(genotypes, "character"))

# observed contingency table for SNP rs630969

table(phenotype,genotypes.df$rs630969,

dnn=c("phenotype","genotype")) # dnn dimension names of table

**A.** Chi-Square Test. The chi-square test compares the observed and expected SNP x Phenotype contingency tables. The expected table shows how the subjects would be distributed in the SNP x Phenotype cells if there were no enrichment of genotypes in one of the phenotype groups. A difference suggests the SNP may increase susceptibility to the disease. We will use fisher.test, an implementation of Fisher’s exact test, used when table cells are sparse.

Use the following code to create a list of contingency tables for all SNPs with sapply.

**3.** Fill in the first blank and show the observed contingency table for SNP rs634228 (hint: type observed.tables.list at the interpreter). Then using test.table fill in the remaining blanks of code to calculate the values for the genotype margins (genoMarg.vec), the phenotype margins (phenoMarg.vec), and the total number of subjects (totalSubj). Hint: use functions **rowSums**, **colSums** and **sum**. Fill the results into the **Observed** Excel Table below.

# creates list of observed contingency tables for all SNPs

# sapply acts on each column of genotypes.df

observed.tables.list <- sapply(genotypes.df, function(x) table(phenotype,x,dnn=c("phenotype","genotype")))

test.table <- observed.tables.list$rs634228 # grab one table

genoMarg.vec <- colSums(test.table) # margin vector

phenoMarg.vec <- rowSums(test.table) # margin vector

totalSubj <- sum(genoMarg.vec) # total subjects

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Observed** | A/A | A/B | B/B |  |
| Case: | 29 | 12 | 0 | 41 |
| Control: | 37 | 10 | 1 | 48 |
|  | 66 | 22 | 1 | 89 |

**4.** Once you create the observed matrix, use the formulas to fill in the **Expected** table below. Fill in table as fractions. Then check your answer with this code:

expect.test <- outer(phenoMarg.vec,genoMarg.vec/totalSubj,'\*')

[out] A/A A/B B/B

0 30.40449 10.13483 0.4606742

1 35.59551 11.86517 0.5393258

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Expected** | A/A | A/B | B/B |  |
| Case: | 66\*(41/89) | 22\*(41/89) | 1\*41/89 | 41 |
| Control: | 66\*48/89 | 22\*48/89 | 1\*48/89 | 48 |
|  | 66 | 22 | 1 | 89 |

Given the observed and expected tables, the chi-square is calculated by the following equation, but some of the cells are sparse or empty, which leads to numerical problems.



The Fisher exact test uses a different approach that is more appropriate for sparse cells. The variable observed.tables.list, computed above, is a list of contingency tables for each SNP. Use the code below to sapply the fisher.test to each table.

**5.** Show the table of the results (fish.results).

# Fisher exact test (chi-square test) for all SNPs

**fish\_fn** <- function(i){

cbind(snp.ids[i], fisher.test(observed.tables.list[[i]])$p.value)

}

# apply fisher exact test to all SNPs

fish.df <- data.frame(t(**sapply**(1:ncol(genotypes.df), **fish\_fn**)))

colnames(fish.df) <- c("rs", "p\_value")

# sort SNPs by Fisher exact p-value

library(dplyr)

fish.results <- fish.df %>%

mutate\_at("p\_value", as.character) %>%

mutate\_at("p\_value", as.numeric) %>%

arrange(p\_value)

print(fish.results)

rs p\_value

1 rs7835221 1.578661e-13

2 rs2460911 8.032123e-04

3 rs2460915 3.925298e-03

4 rs6999231 2.096815e-01

5 rs17786052 2.375773e-01

6 rs529983 2.377522e-01

7 rs12156420 2.919794e-01

8 rs17121574 4.033435e-01

9 rs10105623 4.193435e-01

10 rs634228 4.609209e-01

11 rs2460914 7.209905e-01

12 rs607499 7.237858e-01

13 rs17178729 7.271907e-01

14 rs556531 7.476514e-01

15 rs754238 7.830830e-01

16 rs630969 7.961562e-01

17 rs11203962 8.332052e-01

**B.** Logistic regression with genotypes. In previous labs, we applied logistic regression to a categorical (e.g., case/control) outcome with a numeric (e.g., expression) predictor. Logistic regression also works for categorical (e.g., genotype) predictors.

**6.** Show the plot of the data and logistic model (code below). What genotype has the highest susceptibility for the simulated disease?

library(ggplot2)

i<-8

A1<-ex.data$map$allele.1[i]

A2<-ex.data$map$allele.2[i]

geno.labels <- c(paste(”A”,”A”,sep="/"),paste(”A”,”B”,sep="/"),paste(”B”,”B”,sep="/"))

# data from the one SNP

oneSNP.df <- data.frame(cbind(genotypes.df[[i]],as.numeric(phenotype)))

colnames(oneSNP.df) <- c("genotypes","probability")

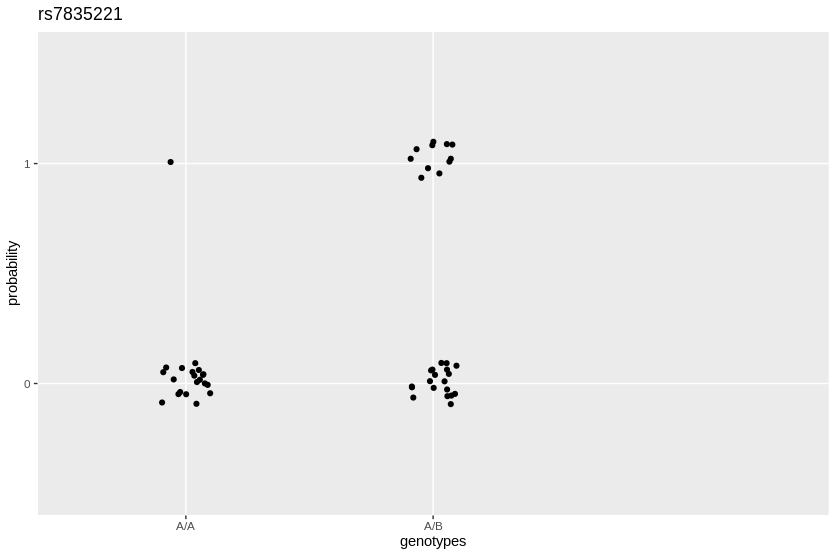
lr.plot <- ggplot(oneSNP.df, aes(x=genotypes, y=probability)) +

geom\_point(position = position\_jitter(w = 0.1, h = 0.1)) +

stat\_smooth(method="glm", method.args = list(family = "binomial")) +

xlim(geno.labels) + ggtitle(snp.ids[i])

print(lr.plot)



**7.** Fill in the blanks in the code below to fit a logistic regression model of the phenotype with the SNP in the *i=8* column. (hint: look at the output of the variable tdlr). A SNP has three genotype states (AA, AB, BB) and hence has three regression beta coefficients.

**8.** What is the rs number of the SNP, what are the genotypes, and what are the regression coefficients for each genotype?

library(broom) # for tidy function. make sure installed

pheno.factor <- factor(phenotype,labels=c(0,1))

i<-8

lr <- glm(pheno.factor~genotypes.df[[i]],family=binomial)

td.lr <- tidy(lr)

pval\_vec <- td.lr$p.value # vector of $p.values from td.lr

coef\_vec <- td.lr$estimate # vector of $estimates

cbind(snp.ids[i], coef\_vec[1], coef\_vec[2], coef\_vec[3], pval\_vec[1], pval\_vec[2], pval\_vec[3])

[out]

[,1] [,2] [,3] [,4] [,5] [,6]

[1,] "rs7835221" "-2.99573226078391" "2.44918855441584" "5.88610401617461" "0.00345833994514128" "0.0249623594060722"

[,7]

[1,] "2.78162616732059e-06"

so rs: rs7835221

genotypes:AA, AB, BB

coef: -2.99573226078391, 2.44918855441584, 5.88610401617461

pvals: 0.00345833994514128, 0.0249623594060722, 2.78162616732059e-06

**9.** Using the code above, fill in the blanks below in the LR.fn function, which will be used in sapply to create a table of logistic regression results for each SNP in the data set. Show the results sorted by the BB homozygous beta coefficient.

2.78162616732059e-06 "rs7835221"

0.366736032162568 "rs607499"

0.524686965956498 "rs630969"

0.736455224784033 "rs10105623"

0.94427390326845 "rs2460914"

0.988502038859224 "rs2460915"

0.990318120879501 "rs17786052"

0.990949929337006 "rs2460911"

0.991349938432283 "rs17121574"

0.991351495823794 "rs11203962"

0.991358016447638 "rs754238"

0.991600015158395 "rs634228"

0.992063090497205 "rs6999231"

0.994401158056065 "rs529983"

<NA> "rs17178729"

<NA> "rs12156420"

<NA> "rs556531"

**10**. How do the SNP rankings compare between logistic regression and the chi-square Fisher test?

The best result is the same by a large margin in both rankings but the rest are somewhat scrambled.

**LR.fn** <- function(i){

lr <- glm(pheno.factor~genotypes.df[[i]],family=binomial)

td.lr <- tidy(lr)

pval\_vec <- td.lr$p.value # vector of $p.values from td.lr

coef\_vec <- td.lr$estimate # vector of $estimates

cbind(snp.ids[i], coef\_vec[1], coef\_vec[2], coef\_vec[3], pval\_vec[1], pval\_vec[2], pval\_vec[3])

}

# apply Logistic Regression model to all SNPs

LRresults.df <- data.frame(t(**sapply**(1:ncol(genotypes.df), **LR.fn**)))

# add column names to results data frame

colnames(LRresults.df) <- c("rs", "AAintercept", "ABcoef", "BBcoef", "AA.pval", "AB.pval", "BB.pval")

# sort LR results by the p-value of the BB homozygous coefficient

# tidy made $p\_value a factor and when you try to convert directly to numeric

# as.numeric turns factors into integer and this messes up sorting

# especially scientific notation

lr.results.sorted <- LRresults.df %>%

mutate\_at("BB.pval", as.character) %>%

mutate\_at("BB.pval", as.numeric) %>%

arrange(BB.pval)

as.matrix(lr.results.sorted %>% pull(rs,BB.pval))

[logistic reg]

2.78162616732059e-06 "rs7835221"

0.366736032162568 "rs607499"

0.524686965956498 "rs630969"

0.736455224784033 "rs10105623"

0.94427390326845 "rs2460914"

0.988502038859224 "rs2460915"

0.990318120879501 "rs17786052"

0.990949929337006 "rs2460911"

0.991349938432283 "rs17121574"

0.991351495823794 "rs11203962"

0.991358016447638 "rs754238"

0.991600015158395 "rs634228"

0.992063090497205 "rs6999231"

0.994401158056065 "rs529983"

<NA> "rs17178729"

<NA> "rs12156420"

<NA> "rs556531"

[chi-square]

rs p\_value

1 rs7835221 1.578661e-13

2 rs2460911 8.032123e-04

3 rs2460915 3.925298e-03

4 rs6999231 2.096815e-01

5 rs17786052 2.375773e-01

6 rs529983 2.377522e-01

7 rs12156420 2.919794e-01

8 rs17121574 4.033435e-01

9 rs10105623 4.193435e-01

10 rs634228 4.609209e-01

11 rs2460914 7.209905e-01

12 rs607499 7.237858e-01

13 rs17178729 7.271907e-01

14 rs556531 7.476514e-01

15 rs754238 7.830830e-01

16 rs630969 7.961562e-01

17 rs11203962 8.332052e-01