Julie Garcia
February 20, 2018

Lab 5
Advanced Genomics and Genetics Analyses

In this lab, we will be analyzing expression quantitative trait loci (eQTL) data. The data matrix is taken from a study where the investigators are profiling both whole genome transcripts and genome-wide SNPs on normal healthy controls and subjects diagnosed with bipolar disorder. The specimen taken for measurement is post mortem brain tissue from the subjects.

We will be focusing on the issues of data manipulation and analysis that go into computing eQTLs between gene expression microarray and SNP array data. There are 83 subjects in each data matrix (i.e. expression matrix and pedigree/map files). The differential expression analysis has already been conducted for genes that discriminate between bipolar and control subjects. We have identified the gene DICER1 as a top candidate for being differentially expressed between the two conditions. Now we want to identify any SNPs, both trans (limited to chromosome 14) and cis, that may be correlated with this gene.

1.) Get the eqtl_files.zip file from the course website and extract all files. Read in all files but the demographic file into R, since we will not be working with this one for this lab. For the pedigree file, take note that it is space delimited and not tab delimited. There are no headers on the ped or map files.

```
map = read.table("C:\\TEMP\\datasets\\bp_ct_83_subjects.map")
map
ped = read.table("C:\\TEMP\\datasets\\bp_ct_83_subjects.ped")
ped
data = read.table("C:\\TEMP\\datasets\\bp_ct_u133A.txt")
data
```

2.) The Affymetrix probe ID for DICER1 is below, as are the SNPs within a 100 kb distance of the DICER1 transcript. Read them into R and find the positions of these SNPs in the map file. You will need to find the actual SNPs in the pedigree file. To do this, use the formula:

Ped position2 = (Map position*2)+6
Ped position1 = Ped position2 – 1

```
#DICER1 gene and SNPs cis to DICER1
probe <- "216280_s_at"
snps <- c("SNP_A-2240938","SNP_A-2249234","SNP_A-2120212","SNP_A-
        1864785","SNP_A-1944204",
"SNP_A-2101022","SNP_A-2192181","SNP_A-4296300","SNP_A-
1961028","SNP_A-2215249","SNP_A-2172180","SNP_A-
1945117","SNP_A-1941491")
```

```
probe <- "216280_s_at"
snps <- c("SNP_A-2240938","SNP_A-2249234","SNP_A-2120212","SNP_A-
1864785","SNP_A-1944204",
          "SNP_A-2101022","SNP_A-2192181","SNP_A-4296300","SNP_A-
1961028","SNP_A-2215249",
          "SNP_A-2172180","SNP_A-1945117","SNP_A-1941491")

dicer1 <- data[probe,]
dicer1


dicer.snps <- map[map$V2 %in% snps,]
dicer.snps
dicer.snps.pos <- rownames(dicer.snps)
dicer.snps.pos
```

3.) Now, subset the pedigree file by these positions of the 13 cis-acting SNPs and apply the following function below to convert the SNPs from a 2 allele genotype code to a single 1 number genotype code (i.e. 11=0, 12/21=1, 22=2, and 00=NA). Hint: use the apply statement on the pedigree file that you just subset.

```
recode <- function(x) {
    x <- as.numeric(x)
    x1 <- seq(1,length(x),by=2)
    x2 <- seq(2,length(x),by=2)
    geno <- paste(x[x1],x[x2],sep="")
    geno[geno=="00"] <- NA
    geno[geno=="11"] <- 0
    geno[geno=="12" | geno=="21"] <- 1
    geno[geno=="22"] <- 2
    geno
}
```

```
m <- matrix(0, ncol = 0, nrow = dim(ped)[[1]])
dicer.snps.ped <- data.frame(m)
for (map.pos in dicer.snps.pos) {
  ped.pos2 <- as.numeric(map.pos)*2+6
  ped.pos1 <- ped.pos2 -1
  colname <- paste("V",as.character(ped.pos1),sep="")
  dicer.snps.ped <- cbind(dicer.snps.ped, ped[,colname])
}

colnames(dicer.snps.ped) <- dicer.snps$V2
dicer.snps.ped
```

```
recode <- function(x) {
  x <- as.numeric(x)
  x1 <- seq(1,length(x),by=2)
  x2 <- seq(2,length(x),by=2)
  geno <- paste(x[x1],x[x2],sep="")
  geno[geno=="00"] <- NA
  geno[geno=="11"] <- 0
  geno[geno=="12" | geno=="21"] <- 1
  geno[geno=="22"] <- 2
  geno
}


dicer.snps.ped.recoded <- apply(dicer.snps.ped,2,recode)
```

4.) Calculate a linear regression model between the gene expression values for DICER1 (using the probe variable and the gene expression matrix) and the single numeric coded genotypes for the 13 cis-acting SNPs. Hint: use the lin.mod() function below with an apply statement.

```
lin.mod <- function(x,y) {
        x <- as.numeric(x)
        dat <- data.frame(y,x)
        out <- lm(y~x,data=dat)
        outx <- summary(out)
        return(outx$coefficients["x",4])
}
```

```
lin.mod <- function(x,y) {
  x <- as.numeric(x)
  dat <- data.frame(y,x)
  out <- lm(y~x,data=dat)
  outx <- summary(out)
  return(outx$coefficients["x",4])
}

pvs <- apply(dicer.snps.ped,2,lin.mod,y=as.numeric(dicer1))
pvs
```

5.) Which of these 13 SNPs has the lowest p-value? What is the p-value?

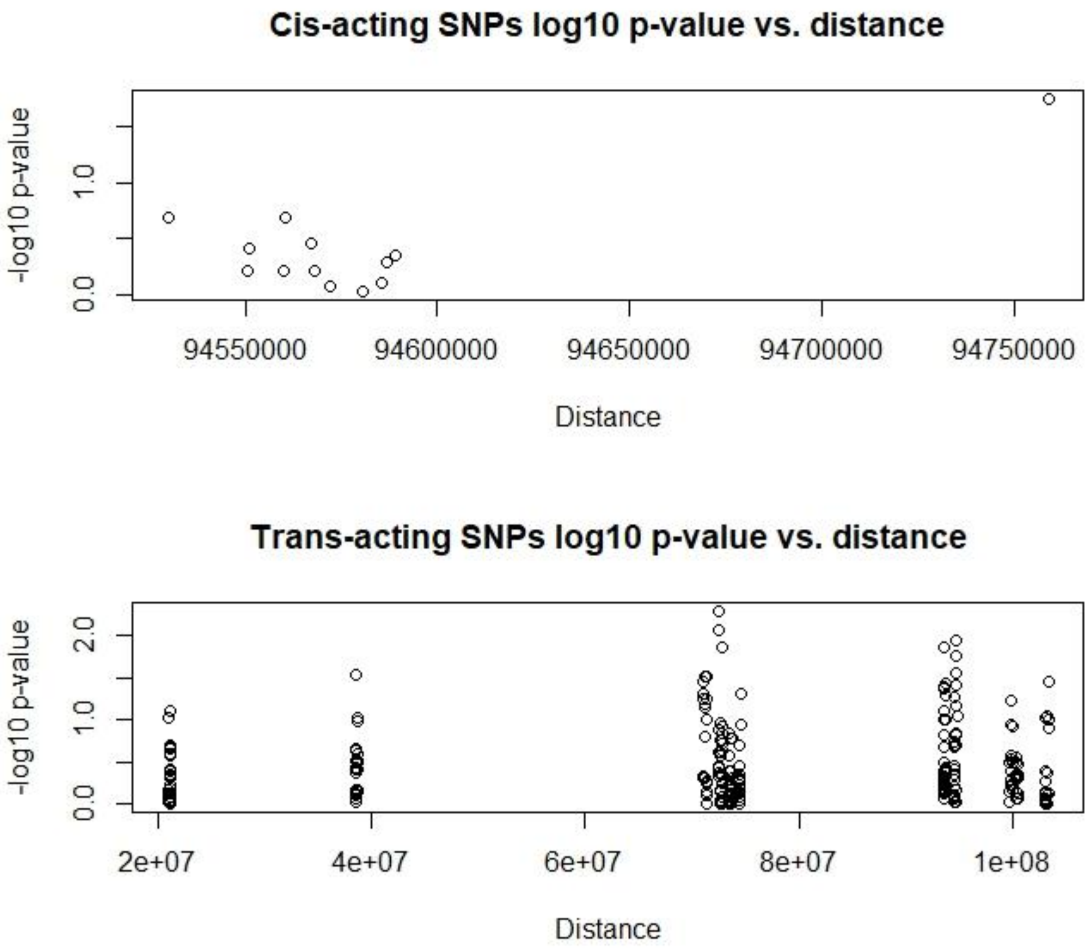
```
The lowest p-value is SNP_A-1941491 at 0.01758504.
```

6.) Now extract all of the remaining SNPs ***only*** on chromosome 14. These are the trans-acting SNPs. Go back to questions #2-4 and recalculate the eQTL p-values, but this time, you will be using 303 trans-acting SNPs, instead of the 13 cis-acting SNPs to DICER1. Hint: use setdiff() to get the remaining SNPs on chromosome 14, different from the 13 SNPs, after you have subset the map file. Then use these SNP names with the match() function on the original map file.

```
dicer.snps.trans <- map[-(map$V2 %in% snps),]
dicer.snps.trans.14 <- dicer.snps.trans[dicer.snps.trans$V1 ==
14,]
dicer.snps.trans.14

dicer.snps.pos.trans <- rownames(dicer.snps.trans.14)
dicer.snps.pos.trans

m <- matrix(0, ncol = 0, nrow = dim(ped)[[1]])
dicer.snps.trans.ped <- data.frame(m)
for (map.pos in dicer.snps.pos.trans) {
  ped.pos2 <- as.numeric(map.pos)*2+6
  ped.pos1 <- ped.pos2 -1
  colname <- paste("V",as.character(ped.pos1),sep="")
  dicer.snps.trans.ped <- cbind(dicer.snps.trans.ped,
ped[,colname])
}

dim(dicer.snps.trans.ped)
colnames(dicer.snps.trans.ped) <- dicer.snps.trans.14$V2
dicer.snps.trans.ped

dicer.snps.trans.ped.recoded <-
apply(dicer.snps.trans.ped,2,recode)
dicer.snps.trans.ped.recoded

pvs.trans <-
apply(dicer.snps.trans.ped,2,lin.mod,y=as.numeric(dicer1))
pvs.trans
```

7.) Use match() on the original map file and identify the physical distances for the 13 cis-acting and 303 trans-acting SNPs. Then plot 2 plots (use the par(mfrow=c(2,1)) function): one with the –log10(p-value) vs. physical distance for the cis-acting SNPs and the other for the trans-acting SNPs.

```
cis.dist <- map[map$V2 %in% colnames(dicer.snps.ped),]$V4
cis.dist
cis.pvalue <- -log10(pvs)

trans.dist <- map[map$V2 %in% colnames(dicer.snps.trans.ped),]$V4
trans.dist
trans.pvalue <- -log10(pvs.trans)

par(mfrow=c(2,1))
plot(cis.dist, cis.pvalue, main="Cis-acting SNPs log10 p-value 
vs. distance",
     xlab="Distance", ylab="-log10 p-value")
plot(trans.dist, trans.pvalue, main="Trans-acting SNPs log10 p-
value vs. distance",
     xlab="Distance", ylab="-log10 p-value")
```


Cis-acting SNPs log10 p-value vs. distance
-log10 p-value
Distance
Trans-acting SNPs log10 p-value vs. distance
-log10 p-value
2e+07
4e+07
6e+07
8e+07
1e+08
Distance