BIOS 7659 Homework 3

Tim Vigers

11 October 2020

# 1. T Statistics

Read in the data:

array <- read.table("./hw3data/hw3arraydata.txt")  
gene\_names <- read.table("./hw3data/hw3genenames.txt",  
 blank.lines.skip = FALSE)

## a) Fold Change

For each gene (row), find the mean expression among controls and among the knock out group. Then calculate fold change using :

fc <- apply(array,1,function(x){  
 control = mean(as.numeric(x[1:8]))  
 knockout = mean(as.numeric(x[9:16]))  
 return(control-knockout)  
})

Top 10 genes with largest absolute value of fold change

|  |  |
| --- | --- |
| Gene | log2FC |
| ApoAI,lipid-Img | 4.749247 |
| EST,HighlysimilartoA | 4.572826 |
| CATECHOLO-METHYLTRAN | 2.772249 |
| EST,WeaklysimilartoC | 1.540431 |
| ESTs,Highlysimilarto | 1.514718 |
| est | 1.466135 |
| similartoyeaststerol | 1.432454 |
| ApoCIII,lipid-Img | 1.398874 |
| psoriasis-associated | 1.256714 |
| Cy3RT | -1.193286 |

## b) Standard t test

For each gene, calculate the two-sample independent t-statistic between controls and knockouts, assuming equal variances:

# Tests  
tp <- apply(array,1,function(x){  
 control = as.numeric(x[1:8])  
 knockout = as.numeric(x[9:16])  
 t <- t.test(control,knockout,var.equal = T)  
 return(c(t$statistic,t$p.value))  
})

Top 10 genes with largest t statistic

|  |  |  |  |
| --- | --- | --- | --- |
| Gene | log2FC | T | p value |
| ApoAI,lipid-Img | 4.7492467 | 23.104347 | 0.0000000 |
| EST,WeaklysimilartoC | 1.5404305 | 12.982368 | 0.0000000 |
| EST,HighlysimilartoA | 4.5728257 | 11.762486 | 0.0000000 |
| CATECHOLO-METHYLTRAN | 2.7722489 | 11.759068 | 0.0000000 |
| ApoCIII,lipid-Img | 1.3988735 | 10.430072 | 0.0000001 |
| est | 1.4661354 | 9.087422 | 0.0000003 |
| ESTs,Highlysimilarto | 1.5147176 | 9.018613 | 0.0000003 |
| similartoyeaststerol | 1.4324539 | 7.208906 | 0.0000045 |
| Caspase7,heart-Img | 0.4533114 | 4.578842 | 0.0004294 |
| EST,WeaklysimilartoF | 0.8558850 | 4.434296 | 0.0005662 |

Out of the 6384 genes, 85 were significant at the p < 0.01 level.

## c) Alternative t statistics

### i) Modified t statistic (using the samr package)

y <- ifelse(grepl("c",colnames(array)),1,2)  
x <- as.matrix(array)  
data=list(x=x,y=y,resp.type="Two class unpaired",  
 assay.type = "array",genenames=gene\_names$V1)  
s <- samr(data)

### ii) Moderated t statistic (using the limma package)

First, create the design matrix for limma:

design <- matrix(ncol = 2,nrow = ncol(array))  
colnames(design) <- c("Control","Knockout")  
rownames(design) <- colnames(array)  
design[,1] <- rep(1,nrow(design))  
design[,2] <- ifelse(grepl("k",rownames(design)),1,0)

Fit the model with limma:

fit <- lmFit(array, design)  
eb <- eBayes(fit)  
limma\_res <- topTable(eb,coef = 2)

Top 10 differentially expressed genes (based on the moderated t statistic)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | logFC | AveExpr | t | P.Value | adj.P.Val | B |
| ApoAI,lipid-Img | -4.749247 | 5.773086 | -23.976817 | 0.0000000 | 0.0000000 | 14.9269328 |
| EST,HighlysimilartoA | -4.572826 | 5.959409 | -12.963071 | 0.0000000 | 0.0000005 | 10.8150265 |
| CATECHOLO-METHYLTRAN | -2.772249 | 6.617134 | -12.439908 | 0.0000000 | 0.0000006 | 10.4483231 |
| EST,WeaklysimilartoC | -1.540431 | 6.817930 | -11.749992 | 0.0000000 | 0.0000012 | 9.9246200 |
| ApoCIII,lipid-Img | -1.398874 | 7.081690 | -9.831229 | 0.0000000 | 0.0000157 | 8.1890866 |
| ESTs,Highlysimilarto | -1.514718 | 7.077908 | -9.012972 | 0.0000000 | 0.0000423 | 7.3031534 |
| est | -1.466135 | 6.971799 | -8.999811 | 0.0000000 | 0.0000423 | 7.2881051 |
| similartoyeaststerol | -1.432454 | 6.640370 | -7.440210 | 0.0000007 | 0.0005617 | 5.3097967 |
| EST,WeaklysimilartoF | -0.855885 | 7.517514 | -4.553948 | 0.0002495 | 0.1769590 | 0.5618636 |
|  | -0.549536 | 7.325818 | -3.961031 | 0.0009254 | 0.5284860 | -0.5563623 |

## d) Method comparisons

# P Values and Multiple Testing

## a) Permutation tests

First, find all the possible permutations of group labels (i.e. control vs. knockout) using combinations(16,8,colnames(array)).

combos <- combinations(16,8,colnames(array))  
pvalues <- apply(array,1,function(g){  
 maxt <- t.test(g[grep("c",names(array))],  
 g[grep("k",names(array))])  
 perms <- apply(combos,1,function(c){  
 control <- g[c]  
 knockout <- g[setdiff(names(g),c)]  
 t <- t.test(control,knockout)  
 return(t$statistic)  
 })  
 return(sum(abs(perms)>=abs(maxt$statistic))/length(perms))  
})