Lab #9

By the beginning of the next lab (April 5th), send what you have to [afodor@uncc.edu](mailto:afodor@uncc.edu)

Make sure the text “Lab #9” is in the subject line…

1. We return to our RNA seq dataset of E. Coli genes from mice…
2. Read and normalize the counts table ( “nc101\_scaff\_dataCounts.txt “ into R). For example:

setwd("C:\\somewhere")

myT<-read.table("nc101\_scaff\_dataCounts.txt",sep="\t",header=TRUE)

myTNorm <- myT

for ( i in 2:ncol(myT))

{

colSum = sum(myT[,i])

myTNorm[,i] =myTNorm[,i]/colSum

}

conditions <- factor( c(rep("W02",3),rep("W12",3),rep("W20",5)))

(The first 3 columns are week 2, the next 3 columns are week 12 and the last 5 are week 20).

1. For every row in the spreadsheet, run a one-way ANOVA with three levels (2 weeks, 12 weeks and 20 weeks) to test the null hypothesis that time had no impact on the counts for each gene. Make a histogram of all the p-values. Do they appear uniform? How many genes are significant at a BH FDR p-value of 0.01?

Hint: if i is a row index, the normalized gene counts can be gotten by

as.numeric( myTNorm[i,2:12] )

1. Make a PDF file that shows the normalized counts for each significant gene at a BH FDR p-value <=.01. Your PDF should have 4 graphs per page (see example page below). Each graph should have the name of the gene and the BH adjusted p-value as title. The graphs should be sorted with the most significant genes shown first.

(The tricky part of this assignment is to sort the graphs by significance!)

(See next page for hints)

Hints for the PDF:

1. The lines:

pdf("output.pdf")

par(mfrow=c(2,2))

will cause calls to “plot” to be output to a PDF, 4 to a page. When you are done with the PDF call dev.off()

If your r-code crashed in the middle, call dev.off() before starting it again

1. If “gene” is a vector that holds your data from one row and “conditions” is defined as above then

aFrame <- data.frame( gene ,conditions)

plot( gene ~ conditions)

stripchart(gene ~ conditions, data = aFrame,vertical = TRUE, pch = 21, add=TRUE )

will make a box-plot and plot the data on top of the boxplot (although there are many, many other ways that you might choose to visualize the data).

1. You can format a p-value to only have a limited number of significant digits with format( myLongPValue,digits=3)

This can make for a less cluttered output

1. As you iterate through each row of the spreadsheet, you can collect the index of the row, the name of each gene and the p-values into separate vectors. Then after you are done running the ANOVA on each gene, you can make a data frame with each vector and sort the data frame by p-value. For example:

aFrame <- data.frame( pValues, bugNames ,rowIndex)

aFrame <- aFrame [order(aFrame$pValues),]

aFrame$adjP <- p.adjust(aFrame$pValues, method="BH")

(The next page shows the first page of my PDF for the assignment with the 4 most significant genes)

