STAT416 **HW5** **Solutions**

1. a) (7 points) An interaction between the factors diet and batch would mean that the differences among the diets would be different for the two batches.

```
> setwd("C:\\z\\Courses\\S416\\GEOdata")
> d=read.table("GSE6297_series_matrix.txt",skip=67,nrows=45101,header=T)
> diet=gl(4,6,labels=c("chimp","McD","cafe","mouse"))
> batch=as.factor(rep(c(1,1,1,2,2,2),4))
> geneNames=d[,1]
> d=d[,-1]
> row.names(d)=geneNames
> names(d)=paste(diet,"B",batch,"R",rep(1:3,8),sep="")
> get.int=function(y)
+ {
+   out=lm(y~diet+batch+diet:batch)
+   a=anova(out)
+   a[3,5]
+ }
> int=t(apply(d,1,get.int))

> hist(int)
```

1. b) (7 points) This histogram shows that the number of genes in any p-value range tends to increase with the p-value. The trend is opposite the trend that would be expected if there were interactions between diets and batches. However, the non-uniform shape of the p-value distribution suggests that there could be a problem with the way that we modeled the data. The p-value distribution should be flat (i.e., uniform) if there were no interactions between diets and batches. Often, a histogram like this indicates that one or more important factors are missing from our model. In this case, the data come from mice of several litters, but we don't have information about which mice come from which litter to include in the model. A non-uniform shape is also made possible by dependence among the p-values due to dependence among genes. In this case though, I suspect a missing factor.

1. c) (7 points) The researchers claim of no interaction may be reasonable.

##########

2. (26 points)

```
> get.p=function(y)
+ {
+   out=lm(y~diet+batch)
+   a=anova(out)
+   df=a[3,1]
+   b=coef(out)
+   v=vcov(out)
+   m=c(0,1,0,0,0)
+   tstat=t(m)%*%b/sqrt(t(m)%*%v%*%m)
+   p12=2*(1-pt(abs(tstat),df))
+   m=c(0,0,1,0,0)
```

```
+   tstat=t(m)%*%b/sqrt(t(m)%*%v%*%m)
+   p13=2*(1-pt(abs(tstat),df))
+   m=c(0,0,0,1,0)
+   tstat=t(m)%*%b/sqrt(t(m)%*%v%*%m)
+   p14=2*(1-pt(abs(tstat),df))
+   m=c(0,-1,1,0,0)
+   tstat=t(m)%*%b/sqrt(t(m)%*%v%*%m)
+   p23=2*(1-pt(abs(tstat),df))
+   m=c(0,-1,0,1,0)
+   tstat=t(m)%*%b/sqrt(t(m)%*%v%*%m)
+   p24=2*(1-pt(abs(tstat),df))
+   m=c(0,0,-1,1,0)
+   tstat=t(m)%*%b/sqrt(t(m)%*%v%*%m)
+   p34=2*(1-pt(abs(tstat),df))
+   p=c(a[1:2,5],p12,p13,p14,p23,p24,p34)
+   results=c(p,a[3,3])
+   results
+ }
> results=t(apply(d,1,get.p))
> p=results[,1:8]
> s2=results[,9]

#############################################################################
> source("http://www.public.iastate.edu/~dnett/microarray/multtest.txt")

3. a) (7 points)

> m0=apply(p,2,estimate.m0)
> m0
[1] 37714.29  8572.00 38850.00 38970.00 36033.33 44666.25 43674.74 39920.00

3. b) (7 points)

> qvals=apply(p,2,jabes.q)
> fdrcuts=c(seq(.01,0.05,by=0.01),.1)
> cbind(fdrcuts,NumberOfGenes=apply(outer(qvals[,7],fdrcuts,"<="),2,sum))
     fdrcuts NumberOfGenes
[1,]    0.01           182
[2,]    0.02           259
[3,]    0.03           330
[4,]    0.04           378
[5,]    0.05           430
[6,]    0.10           682

3. c) (7 points)

> out=ub.mix(p[,3])
> out
[1] 0.5808780 0.5774808 0.9606992
> plot.ub.mix(p[,3],out)
```

3.d) (7 points)

```
> ppdes=ppde(p[,3],out)
> sum(ppdes>=0.75)
[1] 1450

#############################################################################
> library(limma)
> design=model.matrix(~diet+batch)
> colnames(design)=c("mu","d2","d3","d4","b2")
>
> fit=lmFit(d,design)
>
> contr.mat=makeContrasts(d4-d2,levels=design)
>
> fit2=contrasts.fit(fit,contr.mat)
>
> fit3=eBayes(fit2)
```

4. a) (6 points)

```
> fit3$df.prior
[1] 3.097738

> fit3$s2.prior
[1] 0.01418209
```

4. b) (6 points)

```
> mean(fit3$sigma^2>fit3$s2.post)
[1] 0.5801202
```

4. c) (6 points) The usual d.f. for residual would be 24-1-3-1=19. This is the number of observations minus the number of free parameters (1 for mu; 3 for d2, d3, d4; and 1 for b2). To 19, we add d0 to get 22.097738.

4. d) (7 points)

```
> limmaq=jabes.q(fit3$p.value[,1])

> cbind(fdrcuts,NumberOfGenes=apply(outer(limmaq,fdrcuts,"<="),2,sum))
     fdrcuts NumberOfGenes
[1,]    0.01           293
[2,]    0.02           390
[3,]    0.03           473
[4,]    0.04           560
[5,]    0.05           617
[6,]    0.10           884
```

Note that this is considerably more genes at each cut off then we found with just the regular linear model approach.