Homework 3 Solutions
BIOS-7659/CPBS-7659

1) T-statistics

Introduction: A study was performed to examine gene expression changes due to the activity of apolipoprotein AI (apoAI). Mice in a control group (n=8) were compared to mice with the apo AI gene knocked out (n=8). The data were generated on a microarray platform, and have been log transformed. We compared several different statistics (fold change and variations of the t-statistic) for assessing differential expression between the groups. Results for each of the statistics are reported below, along with a discussion about the different test statistics.

a) Using fold-change to assess differences, the top ten genes and their associated log 2 fold change (knockout compared to control) are listed in the table below. Most of the top genes show down-regulation in the knock-out mice (negative fold change).

Row Number	Gene	Fold-Change
2149	ApoAI,lipid-Img	-4.75
540	EST,HighlysimilartoA	-4.57
5356	CATECHOLO-METHYLTRAN	-2.77
4139	EST,WeaklysimilartoC	-1.54
2537	ESTs,Highlysimilarto	-1.51
1496	est	-1.47
4941	similartoyeaststerol	-1.43
1739	ApoCIII,lipid-Img	-1.40
1337	psoriasis-associated	-1.26
5986	Cy3RT	1.19

```
Code:
ApoAI = read.table("hw2data/hw2arraydata.txt",  header = T, sep = "\t")
ApoAInames = read.table("hw2data/hw2genenames.txt",  header = F,
blank.lines.skip = FALSE,as.is = TRUE)[,1] #don't forget to skip blank lines
meandiff  = apply(ApoAI, 1, function(x) mean(x[9:16] - mean(x[1:8])))
        #knock-out vs control
top=order(-abs(meandiff))[1:10]  #don't forget to take absolute value
data.frame(ApoAInames[top], meandiff[top])
```

b) There were 75 genes that were significant at the 0.01 level using a two-sided t-test with unequal variances (85 with equal variance). All genes are down-regulated in the knockout compared to the control

Row Number	Gene	T-statistic	P-value
2149	ApoAI,lipid-Img	-23.10	3.59E-10
4139	EST,WeaklysimilartoC	-12.98	7.19E-09
540	EST,HighlysimilartoA	-11.76	1.88E-06
5356	CATECHOLO-METHYLTRAN	-11.76	1.21E-08
1739	ApoCIII,lipid-Img	-10.43	2.00E-06
1496	est	-9.09	3.06E-06
2537	ESTs,Highlysimilarto	-9.02	6.07E-06
4941	similartoyeaststerol	-7.21	1.23E-05
954	Caspase7,heart-Img	-4.58	5.34E-04
947	EST,WeaklysimilartoF	-4.43	7.89E-04

```
Code:
tstat = apply(ApoAI, 1, function(x) t.test(x[1:8],x[9:16])$statistic)
tpval = apply(ApoAI, 1, function(x) t.test(x[1:8],x[9:16])$p.value)
top=order(-abs(tstat))[1:10] #don't forget the absolute value
data.frame(ApoAInames[top], tstat[top], tpval[top]) #assumes unequal variance
sum(tpval<.01)
```

c)

i) For the SAM Modified t-Statistic, 99 genes are significant at the 0.01 level. Similar genes show up as the other analyses.

Row Number	Gene	Modified T-statistic	p-value
2149	ApoAI,lipid-Img	-20.59	1.57E-04
540	EST,HighlysimilartoA	-11.05	1.57E-04
4139	EST,WeaklysimilartoC	-10.72	1.57E-04
5356	CATECHOLO-METHYLTRAN	-10.63	1.57E-04
1739	ApoCIII,lipid-Img	-8.79	1.57E-04
1496	est	-7.87	1.57E-04
2537	ESTs,Highlysimilarto	-7.85	1.57E-04
4941	similartoyeaststerol	-6.40	1.57E-04
947	EST,WeaklysimilartoF	-3.93	3.54E-04
954	Caspase7,heart-Img	-3.65	5.61E-04

```
Code:
library(samr)
data=list(x=ApoAI,y=c(rep(1,8),rep(2,8)), geneid=ApoAInames, genenames =
       ApoAInames, logged2 = TRUE)
samr.obj<-samr(data, resp.type="Two class unpaired", nperms=100)
pv = samr.pvalues.from.perms(samr.obj$tt, samr.obj$ttstar)
top = order(-abs(samr.obj$tt))[1:10]
data.frame(ApoAInames[top], samr.obj$tt[top],pv[top])
sum(pv<.01)
```

ii) The limma Moderated t-statistic identifies 93 genes significant at the 0.01 level. Similar genes show up as the other analyses

Row Number	Gene	Moderated T-statistic	P-value
2149	ApoAI,lipid-Img	-23.976817	4.74E-15
540	EST,HighlysimilartoA	-12.963071	1.56E-10
5356	CATECHOLO-METHYLTRAN	-12.439908	3.05E-10
4139	EST,WeaklysimilartoC	-11.749992	7.61E-10
1739	ApoCIII,lipid-Img	-9.831229	1.23E-08
2537	ESTs,Highlysimilarto	-9.012972	4.54E-08
1496	est	-8.999811	4.64E-08
4941	similartoyeaststerol	-7.44021	7.04E-07
947	EST,WeaklysimilartoF	-4.553949	2.49E-04
5604		-3.961031	9.25E-04

Code:
```
library(limma)
groups <- c(rep(1,8), rep(2,8))
design <- cbind(WT= 1, KOvsWT= groups-1)
lmresults <- lmFit(ApoAI, design)
fit <- eBayes(lmresults)
modt = fit$t[,2]
modp = fit$p.value[,2]
top = order((modp))[1:10]
data.frame(ApoAInames[top], modt[top], modp[top])
sum(modp<.01)
```

d) Discussion: The fold-change and t-statistic in parts a) and b) follow standard calculations. The alternative t-statistics were developed for microarray data to adjust for possibly unstable variance estimates of genes by pooling information across genes. The penalized t-statistic from SAM adds a small penalty value s_o (e.g., 90th percentile of standard error for all genes) to the standard error in the denominator. The moderated t-statistic from limma is based on an Empirical Bayes method where sample standard deviations s_g have been shrunk towards a pooled standard deviation value s_g^*.

The four methods create slightly different top ten lists. Despite a few differences, the top eight genes occur in all lists with some variations in order. In this example, with sample size of 8 per group, the alternative t-statistics do not make a large difference for the top genes.

The gene "ApoAI,lipid-IMG" appears as the top gene in all lists, where it shows decreased expression in the knock out mice. It is expected that this gene will be ranked high because this study is based on strains with the apo AI gene knocked out. Therefore, expression should be relatively higher in the control mice.

2) P-values and Multiple Testing

Introduction: Continuing with the example above, we will use non-parametric testing approaches as alternatives to parametric t-tests. A variety of multiple testing correction methods will be implemented to address the number of genes tested.

a) Using the exact permutation method there were 117 significant genes at the 0.01 level.

```
Code:
library(gtools)

tstat = apply(ApoAI, 1, function(x) t.test(x[1:8],x[9:16])$statistic)
tpval = apply(ApoAI, 1, function(x) t.test(x[1:8],x[9:16])$p.value)
perms= combinations(16,8)
n = dim(perms)[1]

labels = 1:16
perms= combinations(16,8)
sumt= rep(0,dim(ApoAI)[1])
fullmat = matrix(0, dim(ApoAI)[1], n)
for(i in 1:n)
{
  print(i)
  group1 = perms[i,]
  group2 = labels[-group1]
  perm.tstat = apply(ApoAI,1,function(x) t.test(x[group1],x[group2])$statistic)
#  fullmat[,i] = perm.tstat #can save the permutation t-stats, not used
  sumt = sumt +  (abs(tstat) <= abs(perm.tstat))
}
permp = sumt/n
print(sum(permp <0.01))
```

Note: If the p-value is compared instead of the t-statistic, only 115 genes are identified at the 0.01 level. In theory, using the t-statistic or the p-value would be equivalent. But if the unequal variance version of the t-test is applied (Welch’s t-test), then there is a correction to the degrees of freedom. This correction is based on the variances of the two groups, which will vary based on the particular permutation (and may also be non-integers). Most of the degree of freedom values are between 13-14 (14 is the value for the standard t-test in this example), but some are as low as 8. The result of this correction is that using the t-statistic and p-value is not equivalent.

b) A variety of multiple testing correction methods were applied. The Sidak adjustment for multiple testing is Sidp = $1-(1-p)^m$ where m = 6384. The Bonferroni adjustment is the Taylor series expansion estimate of the exact Sidak adjustment, min(m*p, 1).

The Holm Step-Down adjustment requires that the p-values first be ranked. Then Holmp = $\max_{(k=1,\ldots,j)}(\min((m-k+1)*p_{(k)}, 1))$, where m = 6384 and $p_{(k)}$ = the kth ranked p-value. The max function ensures monotonicity of the p-values. The Benjamini-Hochberg adjustment also requires that the p-values first be ranked. Then BHp = $\min_{(k=j,\ldots,m)}(\min((m/k)*p_{(k)}, 1))$, where again m = 6384 and $p_{(k)}$ = the kth ranked p-value and the first min function ensures monotonicity.

Bonferroni, Sidak and Holm all control the family wise error rate (FWER; probability of making one or more type I errors among all tests) and are more conservative than Benjamini-Hochberg, which controls the false discovery rate (FDR; percent of false discoveries among all test rejections). Bonferroni is the most conservative. Sidak (single-step) is less conservative than Bonferroni. Step-down procedures like Holm and Sidak Step-down procedure are less conservative than single step procedures.

In this data set, Bonferroni, Sidak and Holm all perform the same (see Table below) and report the same number of differentially expressed gene. Only the Benjamini-Hochberg FDR method is less conservative and reports additional genes.

Method	**Unequal Variances**	**Equal Variances**
Bonferroni	7	3
Sidak	7	3
Holm Step-Down	7	3
Benjamini-Hochberg FDR	8	8

Code:
```
ps = apply(ApoAI, 1, function(x) t.test(x[9:16],x[1:8])$p.value)
n = length(ps)

#Bonferroni
bonfp = ps * n
bonfp[bonfp > 1] = 1 #no bigger than 1
sum(bonfp<.01)

#Sidak
sidakp = 1- (1-ps)^n
sidakp[sidakp > 1] = 1 #no bigger than 1
sum(sidakp<.01)

#Holm step-down
ordp = order(ps)
sortp = sort(ps)
ind = 1:n
temp = (n-ind+1)*sortp
mintemp = sapply(temp, function(x) min(x,1))
tempholmp = rep(0,n)
for(i in 1:n)
{   tempholmp[i] = max(mintemp[1:i])  }

holmp = rep(0,n)
holmp[ordp] = tempholmp
sum(holmp<.01)
```

```
#Benjamini-Hochberg
ordp = order(ps)
sortp = sort(ps)
temp = (n/ind)*sortp
mintemp = sapply(temp, function(x) min(x,1))
tempBH = rep(0,n)
for(i in 1:n)
{   tempBH[i] = min(mintemp[i:n])  }


BHp = rep(0,n)
BHp[ordp] = tempBH
sum(BHp<.01)
```

c) The q-value is the expected proportion of false positives at a given significance level or cutoff. It can be thought of as a local discovery rate. In this example, 8 genes have q-values less than 0.01.

The null proportion of genes that are not differentially expressed, π_0, is required to calculate the q-value. This proportion is calculated based on the observed p-values, which under the null should follow a uniform distribution. The `qvalue` package uses p-values larger than some threshold to estimate π_0. In this example, the estimate of π_0 is 0.84.

Code:

```
library(qvalue)
qobj <- qvalue(ps)
qs = qobj$qvalues
sum(qs<.01)
summary(qobj)
```