Intro. to Tests for Differential Expression (Part 1)

Utah State University – Spring 2014
STAT 5570: Statistical Bioinformatics
Notes 3.3

References

- Chapters 14 & 23 of Bioconductor Monograph (course text)
- Tusher, Tibshirani, and Chu (2001). PNAS 98(9):5116-5121
- Broberg (2003) Genome Biology 4:R41.
- Smyth (2004). Statistical Applications in Genetics and Molecular Biology 3(1)#3.
- Stevens, Bell, Aston, and White (2010). BMC Bioinformatics 11:281.

Basic idea of differential expression (DE)

- "Observe" gene expression in different conditions – healthy vs. diseased, e.g.
- Decide which genes' expression levels are changing significantly between conditions
- Target those genes – to halt disease, e.g.
- Note: there are far too many ways to test for DE to present here –
  we will just look at major themes
  of most of them, and focus on implementing a few

Miscellaneous statistical issues

- Test each gene individually
  - Dependence structure among genes not well-understood: (co-regulation or co-expression)
  - Ignore coregulation – first, one at a time
- Scale of data
  - Magnitude of change depends on scale
  - In general: log scale is "approximately right"
  - Variance stabilization transformation can help

Simple / Naïve test of DE

- Observe gene expression levels under two conditions

$$Y_{ijk} = \text{log expr. level of gene k in replicate j of "treatment" i}$$

- Calculate: average log fold change

$$\bar{Y}_{i\cdot k} = \text{ave. log expr. for gene k in treatment i}$$

$$LFC_k = \bar{Y}_{2\cdot k} - \bar{Y}_{1\cdot k} = \text{ave. log fold change for gene k}$$

- Make a cut-off: R

$$\text{Gene k is "significant" if } |LFC_k| > R$$

What does naïve test do?

- Estimate degree of differential expression:
  LFC > 0 for "up-regulated" genes
  LFC < 0 for "down-regulated" genes
- Identifies genes with largest <u>observed</u> change
- Ignores: variability
  - cannot really test for "significance"
  - what if larger LFC have large variability?
    - then not necessarily significant

How to take variability into account?

- Build some test statistic on a per-gene basis
- How do we "usually" test for differences between two groups or samples?
  two-sample t-test
- Test statistic:

$$t_k = \frac{\bar{Y}_{2\cdot k} - \bar{Y}_{1\cdot k}}{s_k} = \frac{LFC_k}{s_k}$$

s_k ← pooled SD

How to use this to "test" for DE?

- What is being tested?

  Null: No change for gene k

- Under null, $\underbrace{t_k \sim \text{t dist.}}_{\text{"parametric" assumption}}$ with n_k d.f.
- But what is needed to do this?
  - "Large" sample size
  - Estimate σ_k = "pop. SD" for gene k (example: s_k)

What if we don't have enough?

- Probably don't –
  even dozens of arrays may not suffice
- Two main problems:
  - 1. Estimate σ_k (especially for small sample size)
  - 2. Appropriate sampling distribution of test stat.
- Basic solutions:
  - 1. To estimate σ_k : Pool information across genes
  - 2. For comparison against 'sampling distribution':
    - use parametric assumption on "improved" test stat.
    - use non-parametric methods – resampling / permuting

Ex 1: Significance Analysis of Microarrays (SAM)

- "Relative difference" test statistic – for gene k

$$d_k = \frac{\bar{Y}_{2\cdot k} - \bar{Y}_{1\cdot k}}{s_k + s_0}$$

s_0 = "tuning" parameter to ensure that var. of d_k is indep. of expression

- How to choose tuning parameter:

  Compute CV of d_k as a function of s_k in moving windows across data, and pick s_0 to minimize CV

  (see SAM users guide and technical document for details)

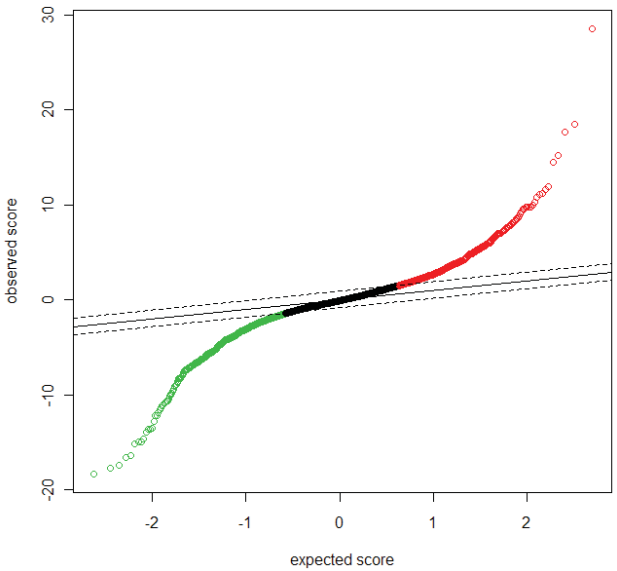
SAM and test for significance

Rather than make distributional assumptions:

- Sort $d_{(1)}, \ldots, d_{(K)}$
- Permute sample labels and re-calculate and sort $d_{p(1)}, \ldots, d_{p(K)}$
- P-value: proportion of permutations where $d_{p(k)}$ is more extreme than $d_{(k)}$
- Let $d_{E(k)}$ be the expected relative difference: average of $d_{p(k)}$ over all permutations
- SAM Plot: plot $d_{E(k)}$ vs. $d_{(k)}$ – look at how far apart they are
- For a given "threshold" Δ, look at genes further away ($d_{E(k)}$ vs. $d_{(k)}$)
  - Generate upper and lower cut-offs for $d_{(k)}$
  - Calculate average (median) # of "significant" genes using same cut-offs on all permutations – this is the estimate of "falsely significant genes"
- Choose Δ based on: desired FDR

SAM plot

delta	# called	median FDR
0.000	4143	0.447
0.011	4121	0.442
0.043	4043	0.434
0.097	3937	0.416
0.173	3652	0.369
0.270	3321	0.304
0.389	2929	0.231
0.529	2520	0.156
0.691	2164	0.093
0.875	1777	0.047
1.080	1465	0.019
1.307	1185	0.006
1.555	966	0.002
1.825	777	0.001
...		



```
### First prepare objects for DE test

# load data
library(affy); library(ALL); data(ALL)

# obtain relevant subset of data; similar Notes 3.2 p. 11
# (filter genes on raw scale, then return to log scale)
library(genefilter); e.mat <- 2^exprs(ALL)
ffun <- filterfun(pOverA(0.20,100))
t.fil <- genefilter(e.mat,ffun)
small.eset <- log2(e.mat[t.fil,])
dim(small.eset) # 4305 genes, 128 arrays

# define comparison to be tested
# first 95 are B-cell, then last 33 are T-cell
T.cell <- c(rep(0,95),rep(1,33))
#  0=B-cell, 1=T-cell
```



```
# Prepare objects for SAM; y must be coded 1/2 (not 0/1)
library(samr)
gn <- rownames(small.eset)
data <- list(x=small.eset, y=(T.cell+1),
  geneid=gn, genenames=gn, logged2=TRUE)

# Call samr - this can take several minutes
samr.obj <- samr(data, resp.type="Two class unpaired",
  nperms=1000, random.seed=1234)

# Choose delta
delta.table <- samr.compute.delta.table(samr.obj)
round(delta.table[,c(1,4,5)],3)

# Visualize results
samr.plot(samr.obj,del=.87)
```



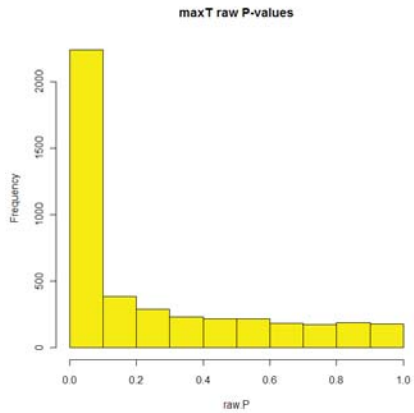
```
# Get names of significant genes
SAM.tab <- samr.compute.siggenes.table(samr.obj, .87,
  data, delta.table)
gn.up <- SAM.tab$genes.up[,3]
gn.dn <- SAM.tab$genes.lo[,3]
gn.SAM <- c(gn.up,gn.dn)
length(gn.SAM) # 1783 genes called sig. by SAM
```


Ex 2: maxT

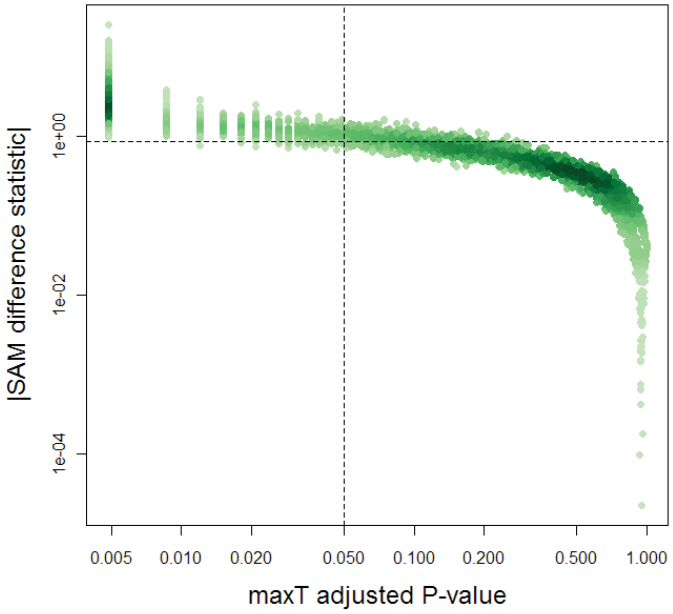
- With larger # of arrays (sample), maybe don't worry so much about:
  "stability" of variability estimate
- Calculate a t-statistic for each gene, then permute sample labels and re-calculate
  - Permutations based on null: sample labels don't matter
  - Welch's: two samples with unequal variances
- P-value for a gene = proportion of permutations that resulted in a more extreme statistic than the original (observed) t-statistic

maxT in R

```
## Look at maxT procedure; data on log scale
library(multtest)
resT <- mt.maxT(small.eset,cl=T.cell,B=1000)

# look at raw P-values
raw.P <- resT$rawp
hist(raw.P,col='yellow',
   main='maxT raw P-values')

# make FDR correction to P-values
adj.p <- p.adjust(raw.P,method='BH')

# find significant genes
t.sig <- adj.p < 0.05
gn.T <- rownames(resT)[t.sig]
length(gn.T) # 1501 genes
```



(cl = classlabel)




```
# Get main results in single data.frame
SAM.frame <- data.frame(gn=names(sort(samr.obj$tt)),
   SAM.diff=abs(sort(samr.obj$tt)-samr.obj$evo))
maxT.frame <- data.frame(gn=rownames(resT), maxT.p=adj.p)
comb <- merge(SAM.frame,maxT.frame)

# Visualize comparison
library(geneplotter)
library(RColorBrewer)
green.ramp <- colorRampPalette(brewer.pal(9,"Greens")[3:9])
dCol <- densCols(log(comb$maxT.p), log(comb$SAM.diff),
  colramp=green.ramp)
plot(comb$maxT.p, comb$SAM.diff, log='xy', pch=16,
     cex.lab=1.5,xlab='maxT adjusted P-value',
     ylab='|SAM difference statistic|',  col=dCol)
abline(v=.05,lty=2)
abline(h=.87,lty=2)
```


A note on permutations

- Both SAM and maxT require permutations
- How many permutations are:
  - Possible?
    - Easy case: compare 2 groups with sizes n_1 and n_2
    - Permuting sample labels is same is rearranging $n=n_1+n_2$ items with n_1 of one kind and n_2 of another:

      Multinomial coefficient: $\frac{(n_1+n_2)!}{n_1!n_2!}$
  - Necessary? – hard to say, but probably: thousands
- In general: do as many as you can afford, especially if total possible is at all manageable