Multiple Testing and Filtering with Gene Expression Data

Utah State University – Fall 2019

Statistical Bioinformatics (Biomedical Big Data)

Notes 5

References

- Chapter 15 of Bioconductor Monograph (course text)
- Benjamini & Hochberg (1995) J. of the Royal Stat. Soc., series B, 57(1):289-300
- Storey & Tibshirani (2003) Proc. of the Natl. Acad. of Science, 100(16):9440-9445
- Hackstadt and Hess (2009). Filtering for Increased Power for Microarray Data Analysis.
- Tuglus and van der Laan (2009). Modified FDR Controlling Procedure for Multi-Stage Analyses. SAGMB 8(1):12
- Tong (2019). Statistical Inference Enables Bad Science; Statistical Thinking Enables Good Science. TAS 73(S1):246-261.

Where are we?

- Up to now:
  - Intro. to gene expression technology
  - Clustering and visualization
    (sometimes using a specific subset of genes)
- Coming up:
  - Testing for differential expression (DE)
    - finding a subset of “significant” genes
  - Annotation and online resources
  - Sequencing
- Here: what to do with DE test results

Differential Expression (DE) tests – basics

- Have 2 or more groups of samples
  ex: healthy, beg. disease, adv. disease

  Null: Gene expressed same in all groups
  Alt.: Gene not expressed same in all groups
  (biological relevance?)
- Result:

  Test Stat.: some "standardized" measure of DE
  – like a t-test, maybe

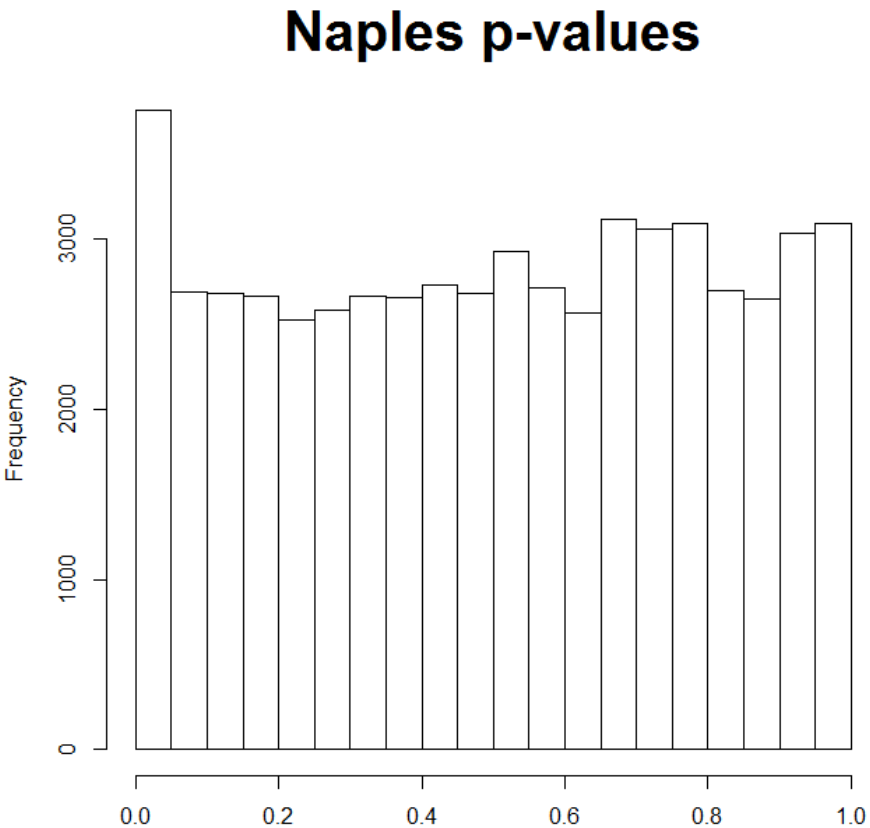
  P-value: some measure of "significance"

Naples Example – simple two-group comparison (Ctrl vs. non-Ctrl)

```
url <- "http://www.stat.usu.edu/jrstevens/bioinf/naplesPvals.csv"
pframe <- read.csv(url, row.names=1)
head(pframe)
```

```
                      pval
ENSG00000000003 0.3984200
ENSG00000000005 0.0109495
ENSG00000000419 0.7563478
ENSG00000000457 0.3742184
ENSG00000000460 0.8755282
ENSG00000000938 0.1393838
```

```
hist(pframe$pval,
  main='Naples p-values',
  xlab=NA, cex.main=2.5)
```



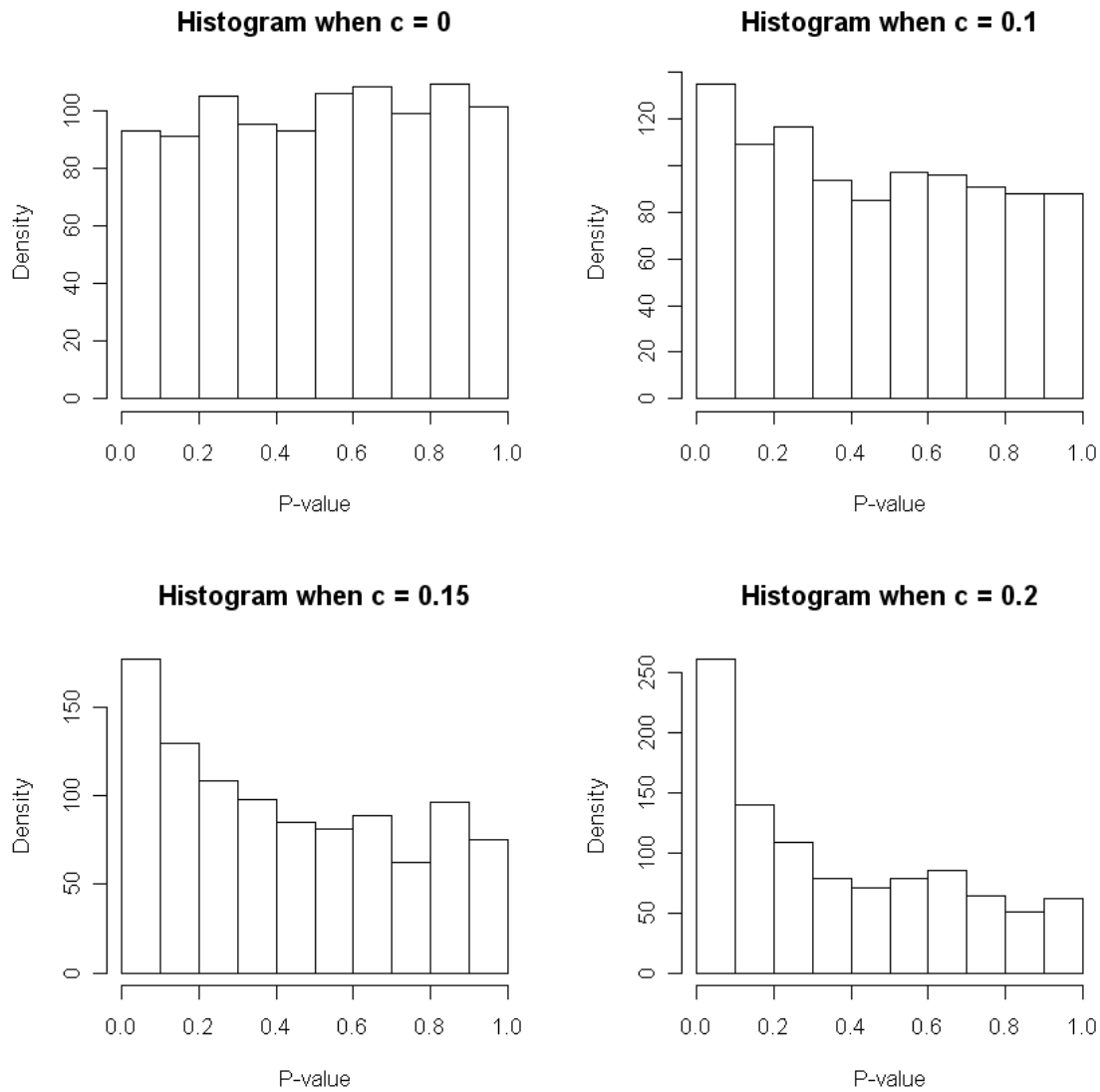
NOTE: We'll come back to how to get p-values like these in Notes 6 & 7).

Significance and P-values

- Usually, "small" P-value → claim significance
- Correct interpretation of P-value from a test of significance:

  "The probability of obtaining a difference at least as extreme as what was observed, just by chance when the null hypothesis is true."

- Consider a t-test of H_0: $\mu_0-\mu_1=0$,
  when in reality, $\mu_0-\mu_1 = c$ (and SD=1 for both pop.)
- What P-values are possible, and how likely are they?



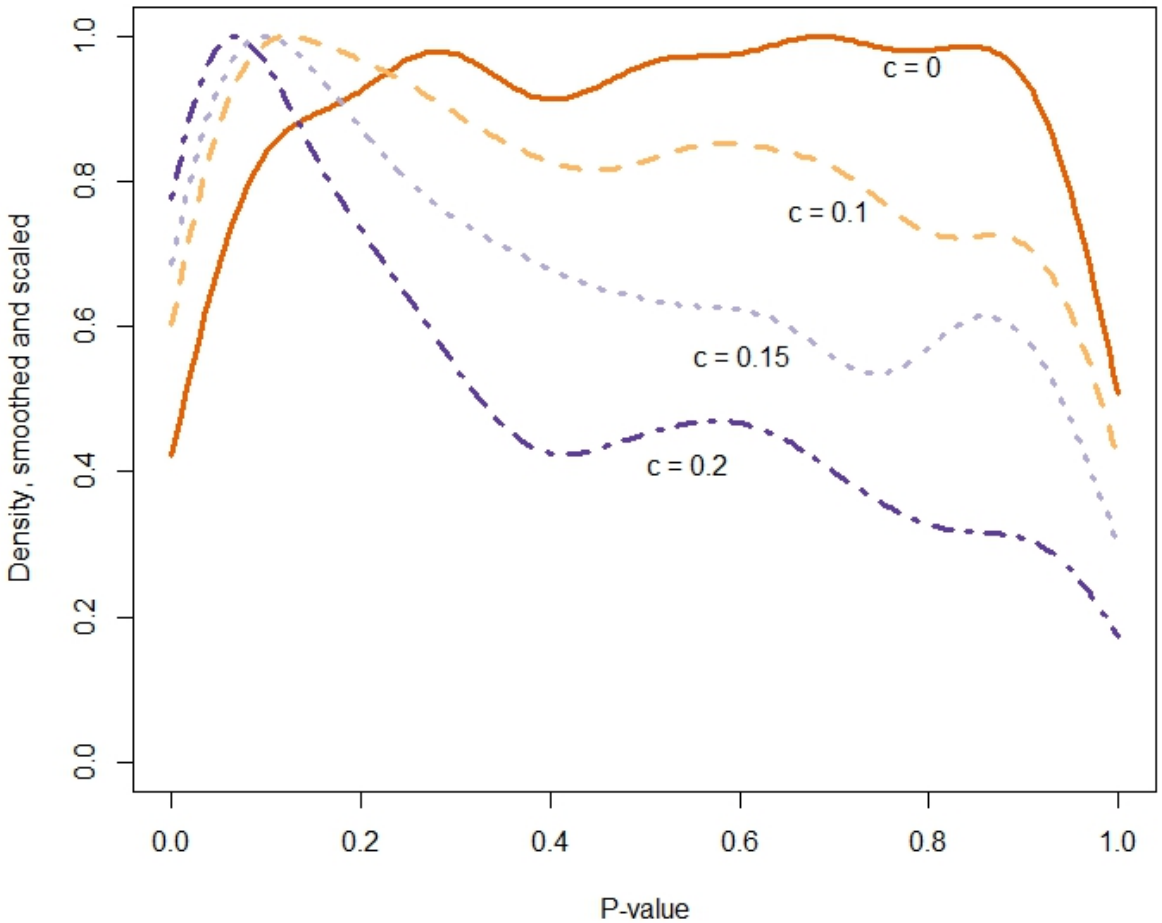
For each value of c, 1000 data sets (think of as 1000 genes) were simulated where two populations are compared, and the "truth" is μ_0-μ_1 = c. For each data set, the t-test evaluates H_0: μ_0-μ_1 = 0 (think of as no change in expression level). The resulting P-values for all data sets are summarized in the histograms.

What's going on here?

```
set.seed(123)
N <- 1000
c.list <- c(0,0.1,0.15,0.2)
k <- length(c.list)
p.mat <- matrix(nrow=N,ncol=length(c.list))
j <- 0
for(c in c.list){
    j <- j+1; p <- 1:N
    for(i in 1:N){
        x <- rnorm(50,mean=c,sd=1)
        y <- rnorm(50,mean=0,sd=1)
        resp <- c(x,y)
        d <- c(rep(0,50),rep(1,50))
        s <- summary(lm(resp~d))$coefficients
        p.mat[i,j] <- s[2,4]} }

par(mfrow=c(2,2))
for(i in 1:k){
   hist(p.mat[,i],xlab='P-value',ylab='Density',
   main=paste('Histogram when c =',c.list[i])) }
```

(Don't worry about this code; it's just here for completeness)

Histograms smoothed and overlayed



Note:

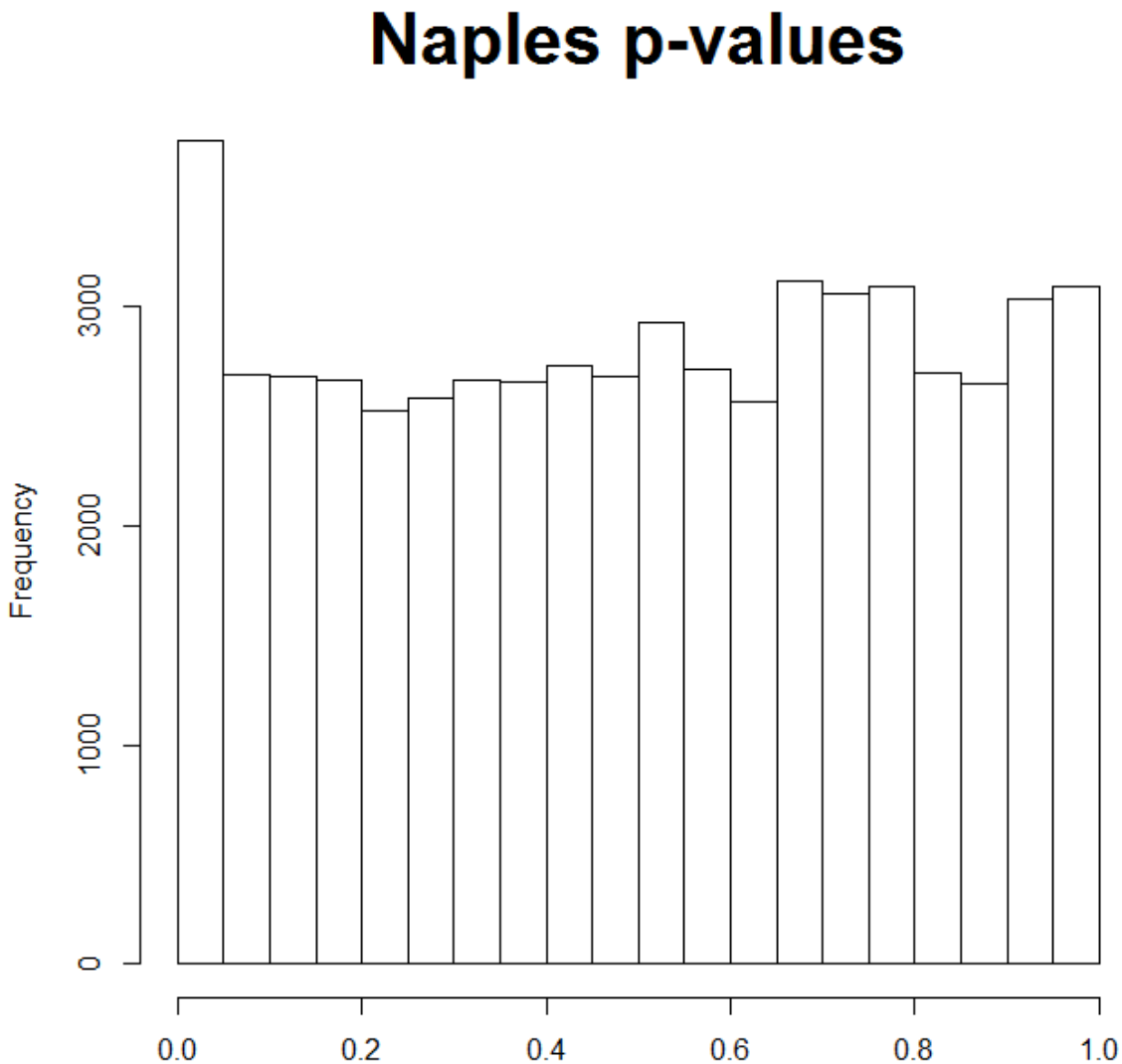
-Even when there is no difference (c=0), very small P-values are possible

-Even for larger differences (c=0.2), very large P-values are possible

-When we look at a histogram of P-values from our test of DE, we have a mixture of these distributions
(because each gene has its own true value for c)

```
n <- 200
x.mat <- y.mat <- matrix(nrow=n,ncol=k)
for(i in 1:k)
    {
     d <- density(p.mat[,i],n=n, from=0, to=1)
     x.mat[,i] <- d$x
     y.mat[,i] <- d$y/max(d$y)
     }
library(RColorBrewer)
cols <- brewer.pal(4, "PuOr")
par(mfrow=c(1,1))
plot(x.mat[,1],y.mat[,1],xlim=c(0,1),type='l',
 lwd=3, xlab='P-value',col=cols[1], ylim=c(0,1),
 ylab='Density, smoothed and scaled')
for(i in 2:k){lines(x.mat[,i],y.mat[,i],col=cols[i],
              lwd=3, lty=i)}
legend(0.7,1.0,paste('c =',c.list[1]),bty='n')
legend(0.6,0.8,paste('c =',c.list[2]),bty='n')
legend(0.5,0.6,paste('c =',c.list[3]),bty='n')
legend(0.45,0.45,paste('c =',c.list[4]),bty='n')
```

(Don't worry about this code; it's just here for completeness)



Remember, this is a mixture of distributions.

A flat histogram would suggest that there really aren't any DE genes.

The peak near 0 indicates that: some genes are DE.

But which ones?

How to treat these P-values?

- Traditionally, consider some cut-off

  Reject null if P-value < α, for example
  (often $\alpha = 0.05$)

- What does this mean?

  α is the acceptable level of Type I error:
  $\alpha = P(\text{reject null} \mid \text{null is true})$

Multiple testing

- We do this with many (thousands, often) genes simultaneously – say m genes

	Fail to Reject Null	Reject Null	Total Count
Null True	U	V	m_0
Null False	T	S	$m-m_0$
	m-R	R	m

# of Type I errors:

V

# of Type II errors:

T

# of correct "decisions":

U+S

Error rates

- Think of this as a family of m tests or comparisons
- Per-comparison error rate: PCER = $E[V/m]$
- Family-wise error rate: FWER = $P(V \geq 1)$
- What does the α-cutoff mean here?

Testing each hypothesis (gene) at level α guarantees:

$$PCER \leq \alpha$$

- let's look at why

What are P-values, really?

Suppose T is the test stat., and t is the observed T.

$$Pval = P(T > t \mid H_0)$$

Assume H_0 is true. Let F be the cdf of T and f be pdf :

$$F(t) = P(T \le t) = \int_{-\infty}^{t} f(t)dt = 1 - Pval$$

What is the distribution of $Y = F(t)$? Let g be pdf of Y :

$$\frac{dy}{dt} = F'(t) = f(t), \quad g(y) = f(t)\left|\frac{dt}{dy}\right| = f(t)\frac{1}{f(t)} = 1$$

So $Y = 1 - Pval$ is $Uniform[0,1]$.

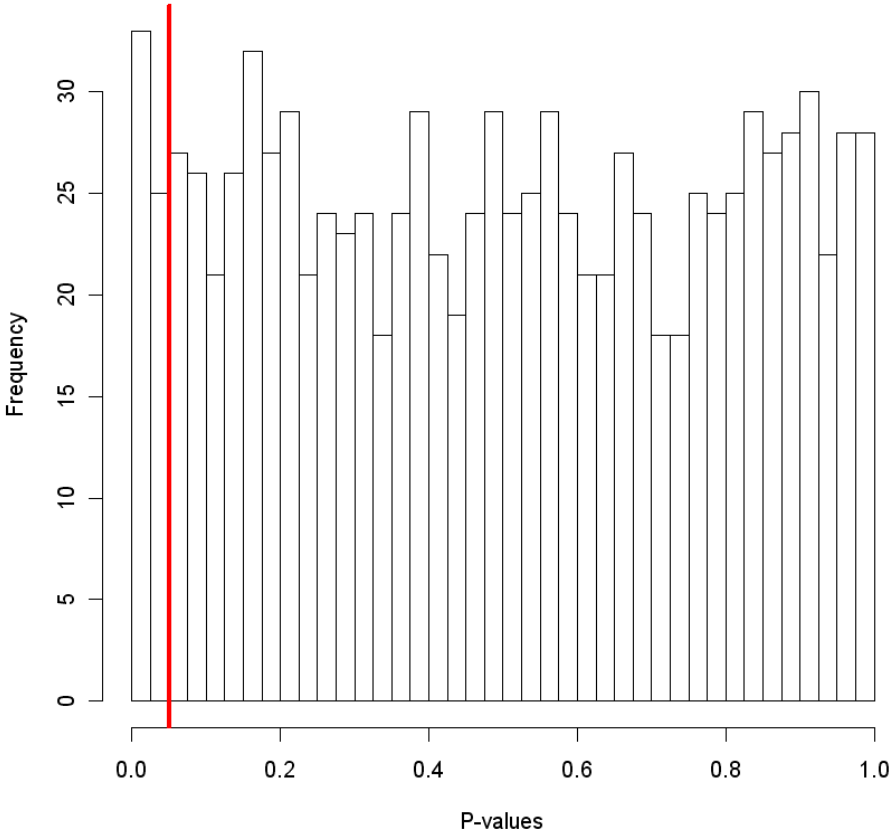
Then when H_0 is true, $Pval \sim U[0,1]$.

P-values and α cut-off

- Suppose null is true for all m genes -
  (so none of the genes are differentially expressed)

- Look at histogram of m=1000 P-values with α=0.05 cut-off -

  about 50 "significant" just by chance
  these can be "expensive" errors




```
set.seed(2); p <- runif(1000)
hist(p,xlab='P-values',main='',
  breaks=c(0:40)/40)
abline(v=0.05,col='red',lwd=3)
```

(Here, V/m ≈ 50/1000 = 0.05.)

How to control this error rate?

Look at controlling the FWER:

Testing each hypothesis (gene) at α/m instead of α guarantees:

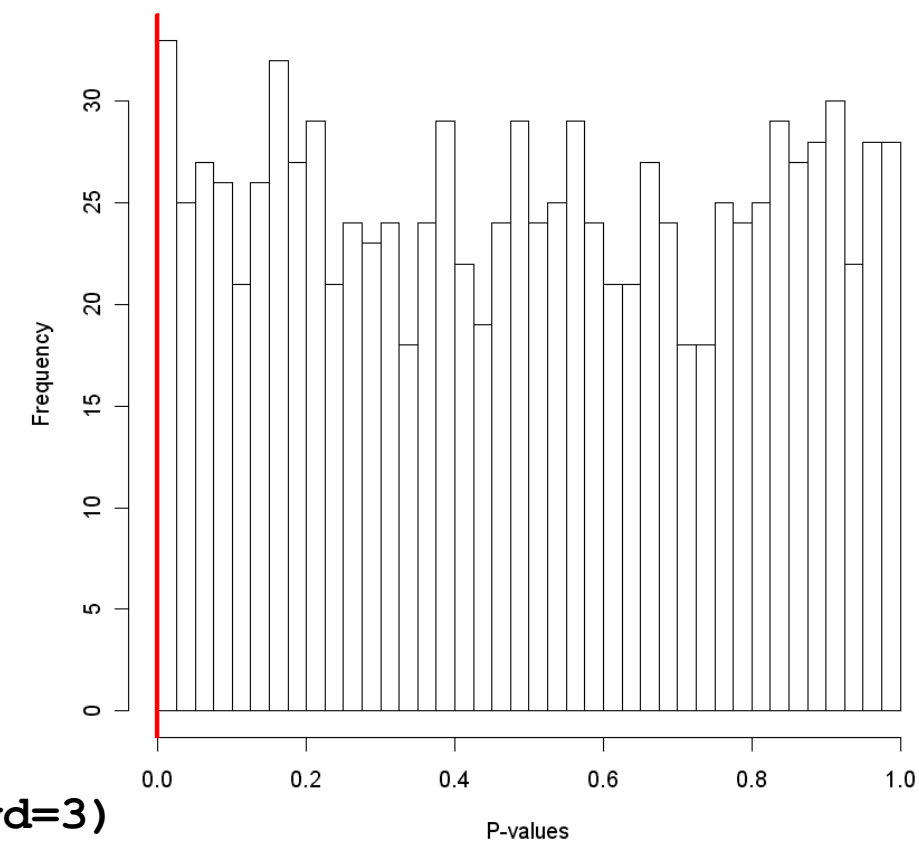
$$FWER \leq \alpha$$

This is called the Bonferroni correction

but -

this is far too conservative for large m

```
hist(p,xlab='P-values',main='',
  breaks=c(0:40)/40)
abline(v=0.05/1000,col='red',lwd=3)
```



A more reasonable approach

- Consider these corrections sequentially:

  Let P_i be the P - value for testing gene i, with null H_i.

  Let $P_{(1)} \leq P_{(2)} \leq \ldots \leq P_{(m)}$ be the ordered P - values.

  Let k be the largest i for which $P_{(i)} \leq \frac{i}{m}\alpha$.

  Reject all $H_{(i)}$ for $i = 1,2,\ldots,k$.

- Then for <u>independent</u> test statistics and for any configuration of false null hypotheses, this procedure guarantees: $E[V/R] \leq \alpha$

What does this mean?

- V = # of “wrongly-rejected” nulls
- R = total # of rejected nulls
- Think of rejected nulls as “discovered” genes of significance
- Then call E[V/R] the FDR
  - False Discovery Rate
- This is the Benjamini-Hochberg FDR correction – sometimes called the marginal FDR correction

Benjamini-Hochberg adjusted P-values

Let $P_{(1)} \le P_{(2)} \le \ldots \le P_{(m)}$ be the ordered P - values.

Let $P^{(adj)}{}_{(i)} = P_{(i)} \cdot \dfrac{m}{i}$.

If any $P^{(adj)}{}_{(i)} > 1$, reset it to 1.

If any $P^{(adj)}{}_{(i)} > P^{(adj)}{}_{(i+1)}$, reset it to $P^{(adj)}{}_{(i+1)}$

(starting at the end of the list, checking backwards).

Then $P^{(adj)}{}_{(1)} \le P^{(adj)}{}_{(2)} \le \ldots \le P^{(adj)}{}_{(m)}$ are

the ordered BH - FDR - adjusted P - values.

An extension: the q-value

- P-value for a gene:

  the probability of observing a test stat. more extreme when null is true

- q-value for a gene:

  the expected proportion of false positives incurred when calling that gene significant

- Compare (with slight abuse of notation):

$$pval = P(T > t \mid H_0 \text{ true}) \;\; ; \;\; qval = P(H_0 \text{ true} \mid T > t)$$

Estimating the q-value

Let $p_{(1)} \le p_{(2)} \le \ldots \le p_{(m)}$ be the ordered P - values.

For $\lambda = 0$ to 0.95 by 0.01 : $\hat{\pi}_0(\lambda) = \dfrac{\#(p_j > \lambda)}{m(1-\lambda)}$.

Let $\hat{f}$ be the natural cubic spline with 3 df of $\hat{\pi}_0(\lambda)$ on λ.

Let $\hat{\pi}_0 = \hat{f}(1)$. ($\pi_0 = m_0/m$ is prop. of genes that are "truly null.")

Calculate $\hat{q}(p_{(m)}) = \hat{\pi}_0 p_{(m)}$.

For $i = m-1, m-2, \ldots, 1$ calculate $\hat{q}(p_{(i)}) = \min\left(\dfrac{\hat{\pi}_0 p_{(i)} m}{i}, \hat{q}(p_{(i+1)})\right)$.

Interpretation

- P-value is a measure of significance in terms of the false positive rate: V/m
- q-value is a measure of significance in terms of the FDR (false discovery rate): E[V/R]

What other adjustments are there?

- More than we could talk about here:

  pFDR = E[V/R | R>0]
  gFWER(k) = P(V ≥ k)
  TPPFP(α) = P(V/R > α)
  maxT – based on ordered test statistics
  minP – based on ordered P-values
  many more … two-step, etc.

  (recall V = # of false disc., R = # of rejected nulls)

- Other ideas: estimating the FDR, estimating the proportion or number of false nulls

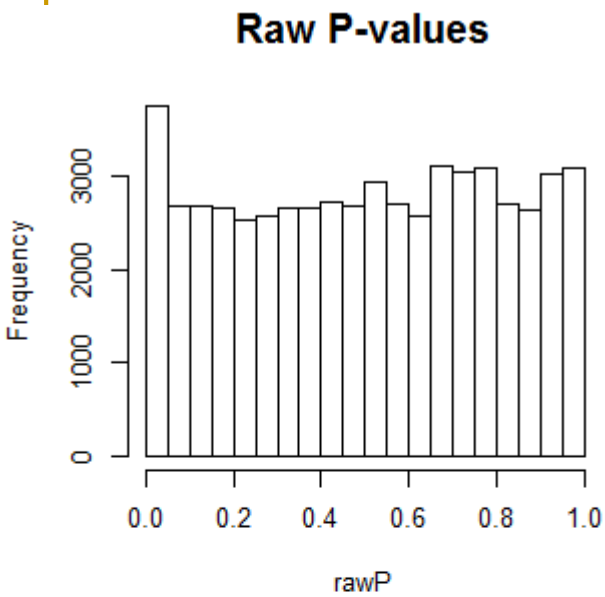
[Aside] Statistical Inference in the 21st Century: A World Beyond $p<0.05$

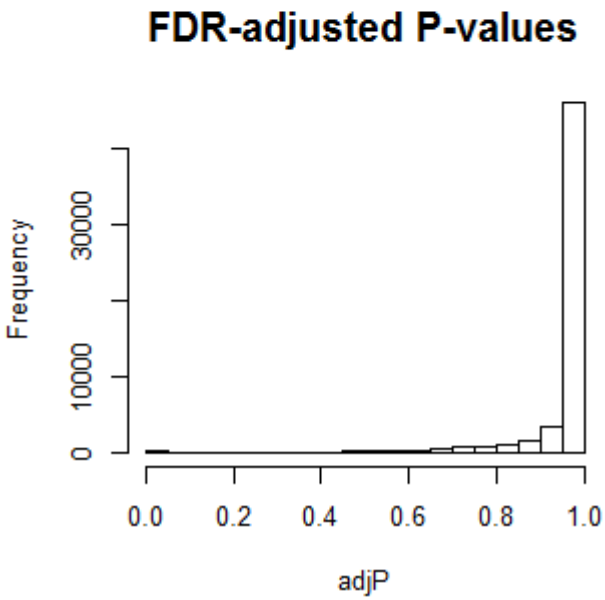
- 2019 special issue of The American Statistician
  → follow-up to 2016 ASA Statement on p-Values and Statistical Significance
  → response to push against NHST framework
  (p-values limited, often misinterpreted)
- "Don't" is not enough;
  Do: accept uncertainty,
  be thoughtful, open, and modest

[Aside] So why use p-values here?

- Should be part of a two-stage process:
  - Transcriptome study → set of "candidate genes" (with specified FDR, E[V/R])
    - Of the R candidate genes, expect no more than V/R to be false positives
  - Validation study
    - Check each of the R genes for DE individually using a more accurate measure (like RT-qPCR) $$$
    - Can report actual FDR
    - Critical to satisfy "many sets of data (MSOD)" vs Isolated Study (Tong 2019)
- Appropriate multiple testing adjustment coupled with real validation (actually do it)

Return to Naples example



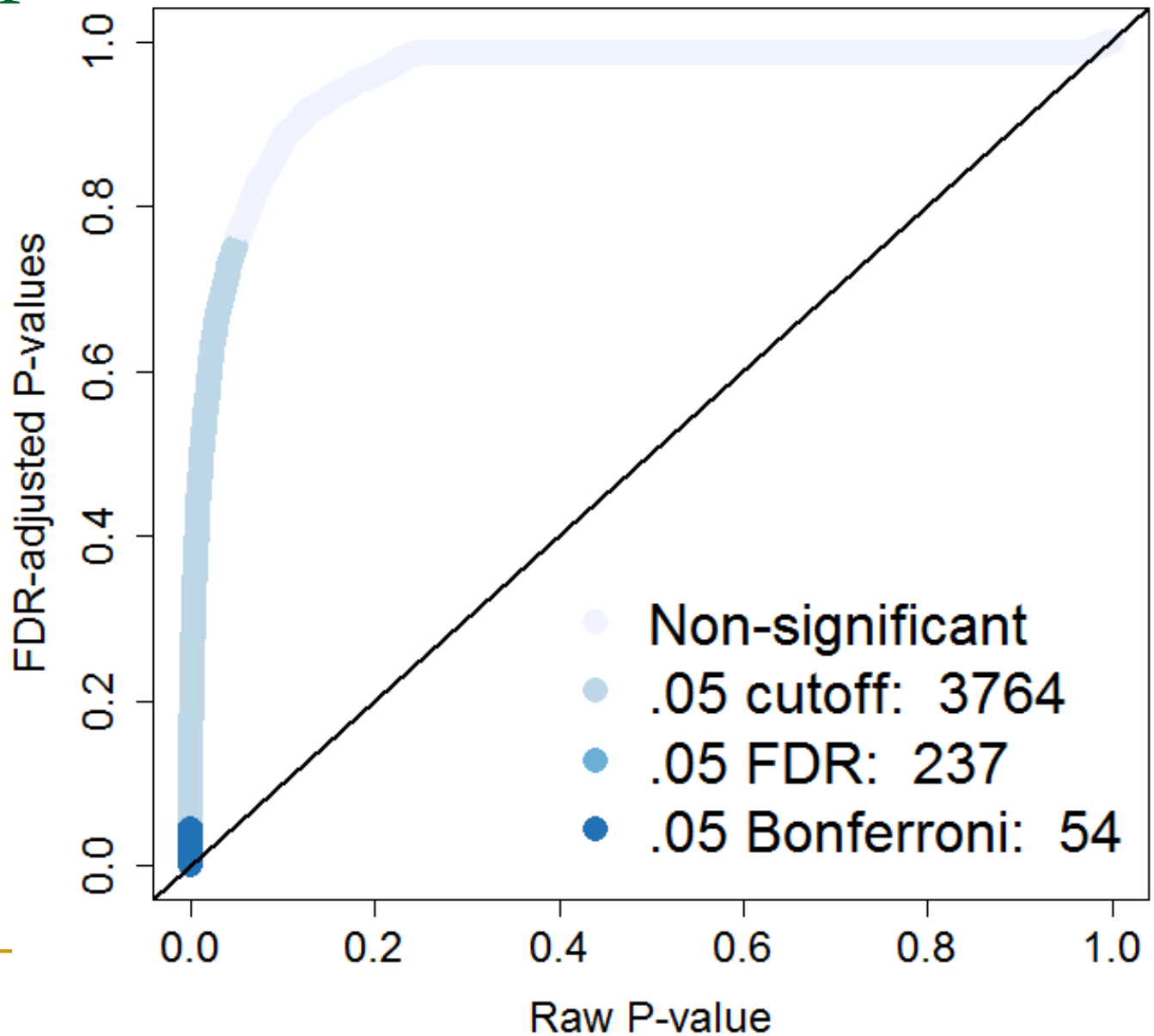



```
# (use Naples results as on slide 5 code)
rawP <- pframe$pval
adjP <- p.adjust(rawP,method='BH')
par(mfcol=c(2,2))
 # NOTE this is different from mfrow
hist(rawP,main='Raw P-values',
  cex.main=1.5)
hist(adjP,main='FDR-adjusted P-values',
  cex.main=1.5)

# See methods automatically available
p.adjust.methods
```

```
[1] "holm"        "hochberg"    "hommel"
[4] "bonferroni" "BH"          "BY"
[7] "fdr"         "none"
```

See “qvalue” package for q-value implementation

Comparison


FDR-adjusted P-values
0.0
0.2
0.4
0.6
0.8
1.0
Raw P-value
0.0
0.2
0.4
0.6
0.8
1.0
Non-significant
.05 cutoff: 3764
.05 FDR: 237
.05 Bonferroni: 54

```
par(mfrow=c(1,1))
library(RColorBrewer)
c.vec <- brewer.pal(4,"Blues")
t.raw <- rawP < 0.05; t.bonf <- rawP < 0.05/length(rawP)
t.FDR <- adjP < 0.05
use.col <- rep(c.vec[1],length(rawP))
use.col[t.raw] <- c.vec[2]; use.col[t.bonf] <- c.vec[3]
use.col[t.FDR] <- c.vec[4]
plot(rawP, adjP, pch=16, cex=2, col=use.col, cex.lab=1.5,
 cex.axis=1.5, xlab='Raw P-value',
 ylab='FDR-adjusted P-values')
abline(0,1, lwd=2)
legend('bottomright',c('Non-significant',
  paste('.05 cutoff: ',sum(t.raw)),
  paste('.05 FDR: ',sum(t.FDR)),
  paste('.05 Bonferroni: ',sum(t.bonf))),
  col=c.vec,pch=16,pt.cex=2,cex=2,bty='n')
```

Which error rate?

- Type I: call gene ‘candidate’ when it’s not
  - PCER / FWER / FDR / etc.
- Type II: fail to identify true candidate
- Relative value (I vs. II) depends on perspective
  - Wasted effort
  - Lost opportunity
- How to reconcile?
  - Sample size → power → low Type II
  - Statistical method → low Type I

One way to increase power: filtering

- Motivation:
  - Relatively few genes should be expressed at any time
  - Relatively few genes should be differentially expressed between conditions
  - Restrict attention to those genes that are relevant "candidates"
- "Non-Specific" Filtering:
  - look only at genes with certain [interesting] properties in "expression values"

Non-specific gene filtering: Variability across all samples

- Genes with large IQR or SD

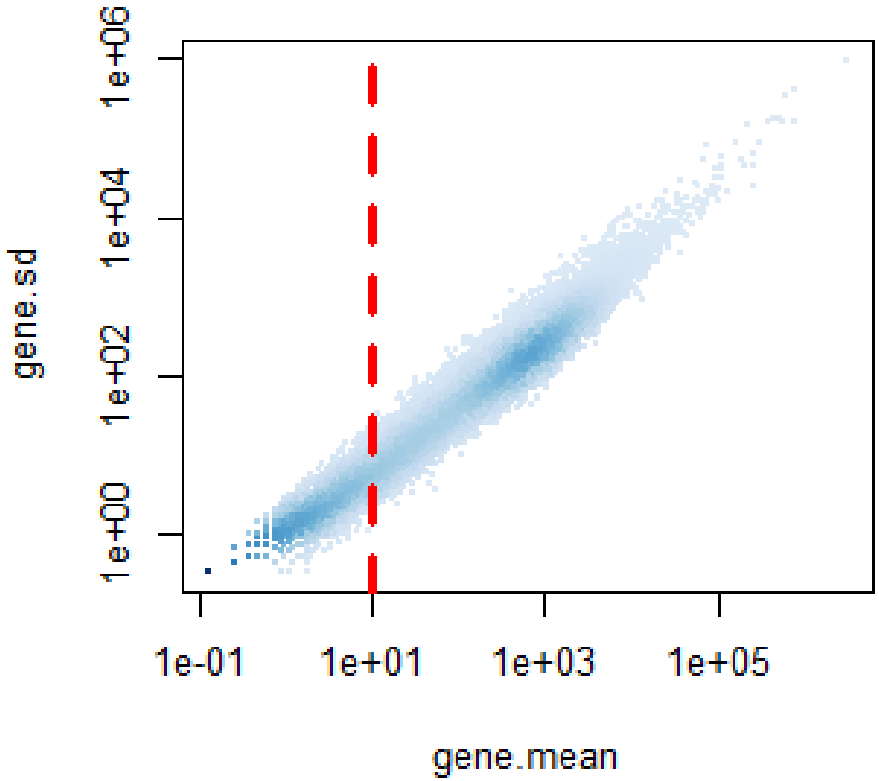
  because: look at genes that actually change expr. levels
  (biological relevance?)

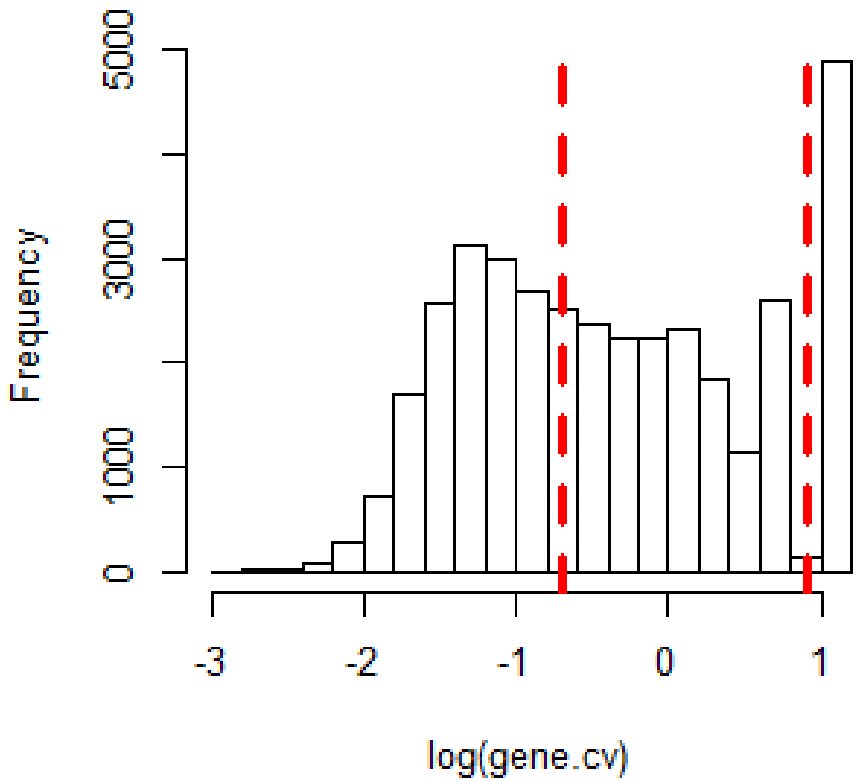
- Compare mean to SD

  because: maybe “tolerate” higher SD for higher expr. levels

- Consider genes with large CV = SD / |mean| = coeff. of var.
  - to balance expr. levels with var. of expr. levels
  - but this can be large just due to - small mean:
    (if gene is “absent” in many samples)
    → look at intensity-based filtering

Example: Look at mean and CV





```
# obtain expression estimates on the UN-LOGGED scale
url <- "http://www.stat.usu.edu/jrstevens/bioinf/naples.csv"
naples <- read.csv(url, row.names=1)
gn <- rownames(naples)
emat <- as.matrix(naples)

# look at mean, sd, & cv for each gene across subjects
gene.mean <- apply(emat,1,mean)
gene.sd <- apply(emat,1,sd)
gene.cv <- gene.sd/gene.mean

# make plots
library(geneplotter); library(RColorBrewer)
blues.ramp <- colorRampPalette(brewer.pal(9,"Blues")[-1])
dCol <- densCols(log(gene.mean),log(gene.sd),
  colramp=blues.ramp)
par(mfrow=c(2,2))
plot(gene.mean,gene.sd,log='xy',col=dCol,pch=16,cex=0.1)
abline(v=10,lwd=3,lty=2,col='red')
hist(log(gene.cv),main=NA)
abline(v=log(.5),lwd=3,lty=2,col='red')
abline(v=log(2.5),lwd=3,lty=2,col='red')
```

Non-specific filtering: Intensity-based

- Expression above some threshold in a certain:
  % of cases (pOverA)

- Expression above some threshold in a certain:
  # of cases (kOverA)

- Choose thresholds based on data, as here:
  - expression above 10 in at least 20% of samples

  and
  - CV between: 0.5 and 2.5

Non-specific filtering and scale

- Non-specific filtering is <u>intensity-based</u>
- Most pre-processing methods return expression estimates on the log-scale
- Before applying non-specific filtering, expression level estimates need to be – "un-logged"

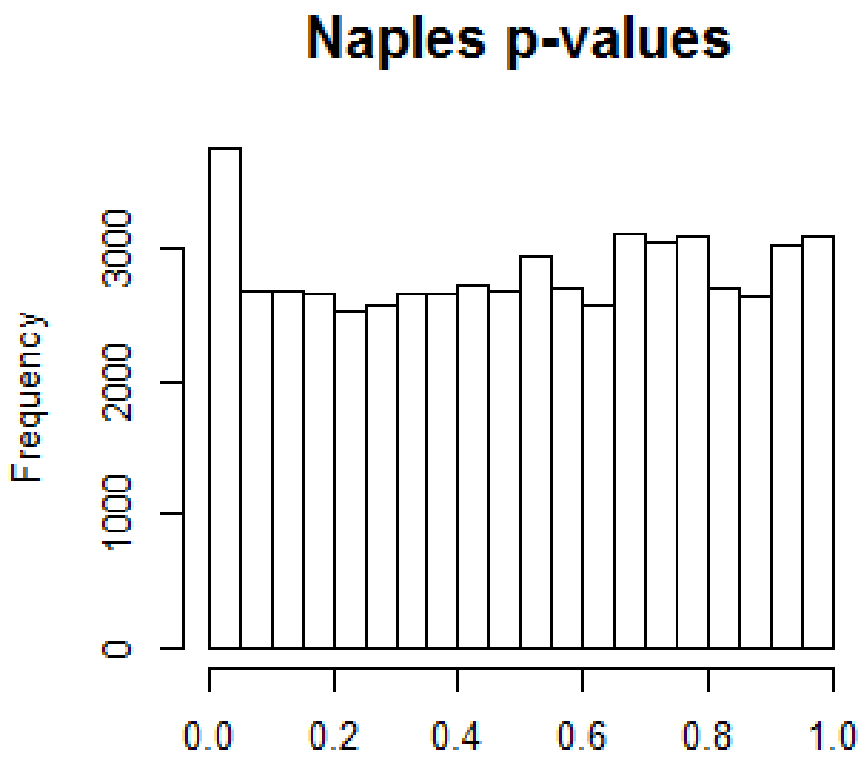
Example: Filter based on CV and pOverA

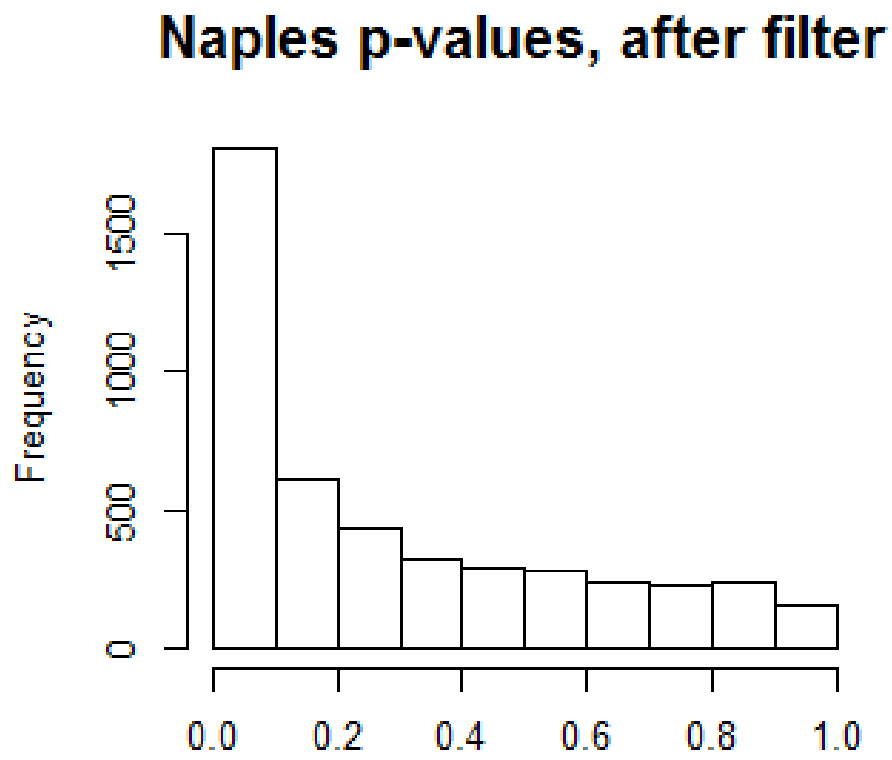
```
# filter: keep genes with cv between .5 and 2.5,
#         and where at least 20% of samples had exprs. > 10
library(genefilter)
ffun <- filterfun(pOverA(0.20,10), cv(0.5,2.5))
t.fil <- genefilter(emat,ffun)
# apply filter, and put expression back on log scale
eset.fil <- emat[t.fil,]

dim(emat)
# 56621  8
dim(eset.fil)
#  4597  8

# find the gene names
gn.keep <- rownames(eset.fil)
```

Then test for DE using <u>only</u> the genes in this eset.fil object. (Here, 4,597 genes instead of all 56,621.)

Visualize filter's effect on power

```
# Use Naples p-values in pframe object from slide 5,
# and gene names passing filter (gn.keep object on slide 37)
t <- is.element(rownames(pframe),gn.keep)
par(mfrow=c(2,2))
hist(pframe$pval, main='Naples p-values', xlab=NA, cex.main=1.5)
hist(pframe$pval[t], main='Naples p-values, after filter',
 xlab=NA, cex.main=1.5)
```





Framework for filter: multi-stage analysis (MSA)

- Stage 1: Apply filter; partition set of raw P-values

p_1	…	p_r	p_{r+1}	…	p_m

filtered out — pass filter

- Commonly-used Stage 2: Apply MCP (like FDR or q-value) <u>adjustment</u> to p-values of tests that pass filter

NA	…	NA	$\underline{p}_{r+1}$	…	$\underline{p}_m$

What if filter is based on rank of p-values? Control error rate in multi-stage: FDR-MSA

- Stage 2a: Reset filtered-out p-values to 1

$p'_1=1$	…	$p'_r=1$	p_{r+1}	…	p_m

pass filter (smaller p-values)

- Stage 2b: Apply FDR <u>adjustment</u> to [reset] p-values of all tests

$\underline{p}'_1$	…	$\underline{p}'_r$	$\underline{p}_{r+1}$	…	$\underline{p}_m$

- <u>When the filter preserves the rank of the p-values</u>, this will appropriately control Type I and II rates

But – when will filter preserve rank of p-values?

Concerns & final notes on filtering

- Can be a bit subjective: which threshold and why?
- Can erroneously eliminate important genes
- Can ignore experimental design
  - Maybe more important than high CV:
    - var. low within groups, but high between
    - but – this could “double dip” test statistic
  - Avoid filtering on anything related to test statistic
    - Double-dipping changes meaning of α
    - Otherwise, need to control error rate in multi-stage analysis
- Can help reduce multiple testing issues (make adjustments less severe) while raising power (should give greater concentration of DE genes)

Summary

- Tests of differential expression
  - Null: “gene is not DE” vs. Alt: “gene is DE”
  - Test Stat. → P-value
- How to treat P-values: uniform random variables
- Multiple comparison procedures
  - simple cut-off → too liberal
  - Bonferroni correction → too conservative
  - FWER
  - FDR and q-values
  - others – we may return to this topic: good 6000-level projects
- Filter (carefully) to improve statistical power