Using limma for Differential Expression

James W. MacDonald
jmacdon@med.umich.edu

BioC 2010
July 29, 2010

Overview

Overall goal is to teach use of limma

- Example analyses
  - colonCA
  - estrogen
- Statistical discussions
  - Linear models
  - Experimental design
  - Design/contrast matrices
  - Multiple comparisons
- Visualization/output of results

Why limma?

- Fit nearly any model
- Technical replicates
- One/two color arrays
- Increased power

Why not limma?

- Complexity
- Reliance on normal theory
- Can't fit linear mixed models
- Can't handle multiple levels of technical replication

Normal analysis workflow

- Import data
- Pre-process
- Fit model(s)
- Make comparisons
- Filter data
- Output results

Load Data

Load data we will use today.

```
> x <- "http://www.umich.edu/~jmacdon/BioC2010.Rdata"
> con <- url(x)
> load(con)
> close(con)
```

If using thumb drive, start R in directory containing BioC2010.Rdata, then

```
> load("BioC2010.Rdata")
> ls()

[1] "colonCA"  "estrogen"
```

colonCA

```
> library(Biobase)
> head(pData(colonCA))

  expNr samp class
1     1   -1     t
2     2    1     n
3     3   -2     t
4     4    2     n
5     5   -3     t
6     6    3     n
```

Simple *t*-test

Assume no pairing

Two common parameterizations

Cell means model

Baseline model

These parameterizations are equivalent

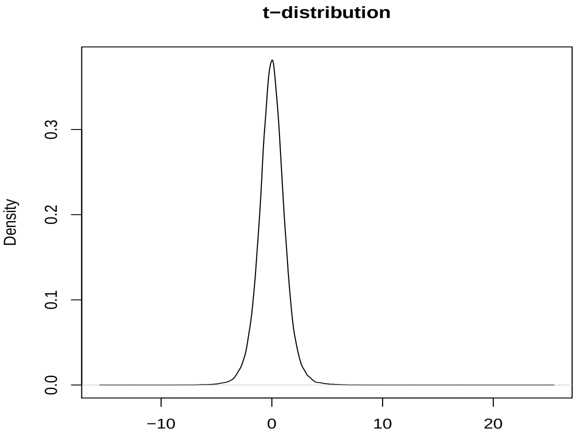
The t-statistic

General form of a t-statistic

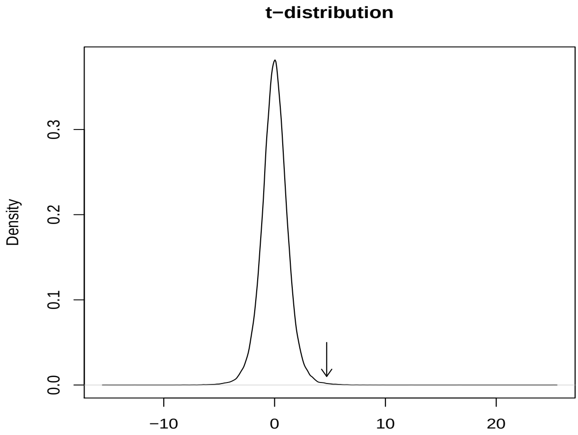
$$t = \frac{\hat{\beta}}{\frac{s}{\sqrt{n}}}$$

- Numerator captures differences
- Denominator acts as 'yardstick' for numerator
- We are testing $\beta = 0$
- Compare to reference distribution to assess significance

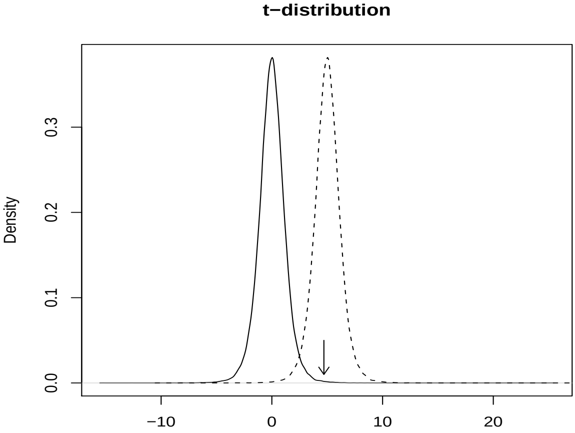
Inference for *t*-statistic



Inference for *t*-statistic



Inference for *t*-statistic



Cell Means Model

$$y_{ij} = \mu_j + \epsilon_{ij}$$

or

$$y_{tumor1} = \mu_{tumor} + \epsilon_{tumor1}$$

$$y_{normal1} = \mu_{normal} + \epsilon_{normal1}$$

or

$$y_{ij} = I\mu_{tumor} + I\mu_{normal} + \epsilon_{ij}$$

$$I \in (0, 1)$$

Cell Means Design Matrix

```
> design <- model.matrix(~0+pData(colonCA)$class)
> colnames(design) <- c("Normal","Tumor")
> head(design)

  Normal Tumor
1      0     1
2      1     0
3      0     1
4      1     0
5      0     1
6      1     0
```

Cell Means Contrast Matrix

Recall *t*-statistic:

$$t = \frac{\hat{\beta}}{\frac{s}{\sqrt{n}}}$$

```
> makeContrasts(Tumor - Normal, levels = design)

         Contrasts
Levels    Tumor - Normal
  Normal              -1
  Tumor                1
```

Fit Cell Means Model

```
> fit <- lmFit(colonCA, design)
> fit <- contrasts.fit(fit, contrast)
```

Baseline Model

$$y_{ij} = \alpha + \tau_j + \epsilon_{ij}$$

or

$$y_{normal1} = \alpha + \epsilon_{normal1}$$

$$y_{tumor1} = \alpha + \tau_1 + \epsilon_{tumor1}$$

or

$$y_{ij} = \alpha + I\tau_j + \epsilon ij$$

$$I \in (0, 1)$$

Baseline Model Design Matrix

```
> tumorvnormal <- pData(colonCA)$class
> design <- model.matrix(~tumorvnormal)
> colnames(design) <- c("Intercept","Tumor-Normal")
> head(design)

  Intercept Tumor-Normal
1         1            1
2         1            0
3         1            1
4         1            0
5         1            1
6         1            0
```

Fit Baseline Model

```
> fit <- lmFit(colonCA, design)
```

Now what?

Sample size

The *t*-statistic (again)

$$t = \frac{\hat{\beta}}{\frac{s}{\sqrt{n}}}$$

Denominator dependent on

- Sample variability
- Number of replicates

Sample variability dependent on

- Number of replicates

eBayes

Fewer replicates increase variability:

- Mathematically
- By chance

eBayes step estimates 'average' variability over all genes and

- Adjusts high variability genes down
- Adjusts low variability genes up

Filter data

Things to consider:

- eBayes needs all genes
- Multiple comparisons problem
- Statistical vs biological significance

Selecting 'Top' Genes

- eBayes/topTable control output by
  - Coefficient of interest
  - Number of genes
  - *p*-value (adjusted)
  - Fold change
- treat/topTreat control output by
  - All of the above
  - Incorporates fold change into computation of *p*-value

Output data

```
> fit2 <- eBayes(fit)
> output <- topTable(fit2, coef = 2)
> ## or
> fit2 <- treat(fit)
> output <- topTreat(fit2, coef = 2)
```

Output

How do the results for eBayes and treat differ?
Check man pages for how to incorporate fold change.
How do the results differ when adding a fold change criterion?
How would one select genes with a FDR of 5%?

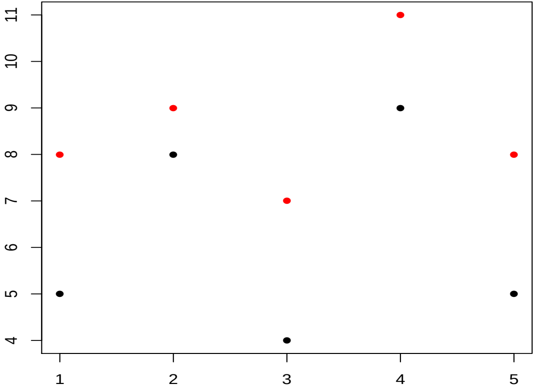
Paired Analysis

Colon cancer data are actually paired:

```
> head(pData(colonCA))
  expNr samp class
1     1   -1     t
2     2    1     n
3     3   -2     t
4     4    2     n
5     5   -3     t
6     6    3     n
```

So how do we handle this aspect?

Paired data



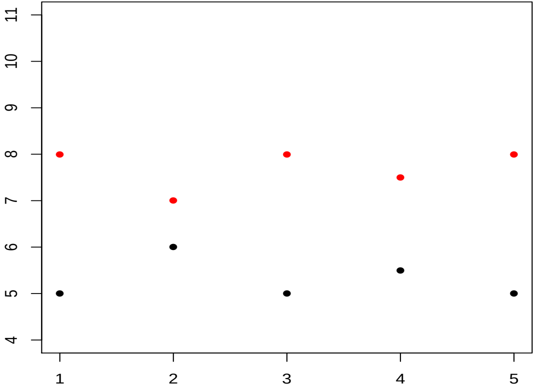
Paired analysis 'by hand'

```
> ind <- pData(colonCA)$samp > 0
> dat <- exprs(colonCA)[,!ind] -
+     exprs(colonCA)[,ind]
> design <- rep(1, ncol(dat))
> fit <- lmFit(dat, design)
> fit.pair1 <- eBayes(fit)
```

Paired analysis using batch term

```
> pair <- factor(abs(pData(colonCA)$samp))
> design <- model.matrix(~tumorvnormal +
+                              pair)
> colnames(design) <- c("Intercept", "Tumor-Normal",
+                         paste("Pair", 2:22, sep=""))
> head(design)[,1:4]
  Intercept Tumor-Normal Pair2
1         1            1     0
2         1            0     0
3         1            1     1
4         1            0     1
5         1            1     0
6         1            0     0
  Pair3
1     0
2     0
```

Paired analysis using batch term



```
> fit <- lmFit(colonCA, design)
> fit.pair2 <- eBayes(fit)
> comp <- cbind(fit.pair1$coef, fit.pair2$coef[,2])
> colnames(comp) <- c("Direct","Batch")
> head(comp)

            Direct Batch
Hsa.3004      0.25  0.25
Hsa.13491     0.14  0.14
Hsa.13491.1   0.22  0.22
Hsa.37254     0.13  0.13
Hsa.541       0.30  0.30
Hsa.20836     0.16  0.16
```

Estrogen Data

```
> pData(estrogen)
             sample
high10-1.cel      1
high10-2.cel      2
high48-1.cel      3
high48-2.cel      4
low10-1.cel       5
low10-2.cel       6
low48-1.cel       7
low48-2.cel       8
```

Comparisons

Comparisons of interest:

- High vs low estrogen
- 10 vs 48 hour incubation
- Interaction

Estrogen Design Matrix

What would a cell means design matrix look like?
How would it be constructed?
Hint: See ?formula
How about the contrasts matrix?

Estrogen Design Matrix I

```
> Level <- factor(rep(c("High","Low"), each=4))
> Time <- factor(rep(c("10","48"), each = 2,
+                    times = 2))
> design <- model.matrix(~0+Level*Time)
```

Estrogen Design Matrix

What would a baseline design matrix look like?
How would it be constructed?
How about the contrasts matrix?

Estrogen Design Matrix II

```
> design <- model.matrix(~Level*Time)
```

Estrogen Differential Expression

```
> fit <- lmFit(estrogen, design)
> fit2 <- eBayes(fit)
```

But now what?

decideTests

topTable for multiple coefficients
Various options to control multiplicity

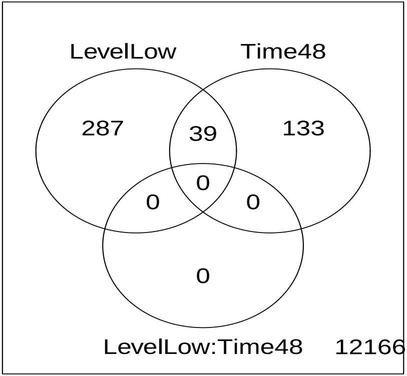
- separate
- global
- hierarchical
- nestedF

decideTests

```
> rslt <- decideTests(fit2[,2:4])
> rslt[1:5,]
          LevelLow Time48
1000_at          0      0
1001_at          0      0
1002_f_at        0      0
1003_s_at        0      0
1004_at          0      0
          LevelLow:Time48
1000_at                 0
1001_at                 0
1002_f_at               0
1003_s_at               0
1004_at                 0
```

vennDiagram

```
> vennDiagram(rslt)
```



GSEA

No significant genes in interaction.

Does that mean we are done?

Consider genes as a set instead of individually.

- geneSetTest
  - Competitive analysis
  - H_0: Our set of genes no more differentially expressed than the remainder of genes on the chip.
- roast/romer
  - Self-contained analysis
  - H_0: Our set of genes is not differentially expressed.

geneSetTest

Two required arguments:

- 'Indicator'vector for genes of interest
- Vector of statistics

```
> ind <- as.numeric(row.names(topTable(fit2,
+                                      coef = 4,
+                                      number = 100,
+                                      p.value = 0.2)
> geneSetTest(ind, fit2$t[,4])
```

```
[1] 2.3e-14
```

roast

Four required arguments:

- 'Indicator' vector for genes of interest
- Matrix of expression values
- Design matrix
- Contrast matrix

```
> roast(ind, exprs(estrogen), design, c(0,0,0,1))

      Active.Prop P.Value
Mixed        1.00   0.003
Up           0.32   0.990
Down         0.68   0.011
```