Analyzing One-Color Data with limma

James W. MacDonald
jmacdon@med.umich.edu

BioC 2008
July 28, 2008

Introduction

Assumptions

- Data are one-channel microarray data
  - Affymetrix
  - Nimblegen
  - Possibly cDNA chip with common reference
- We assume data have been normalized and summarized
- Goal is to make comparisons
  - *t*-tests
  - linear models

limma package

Why limma?

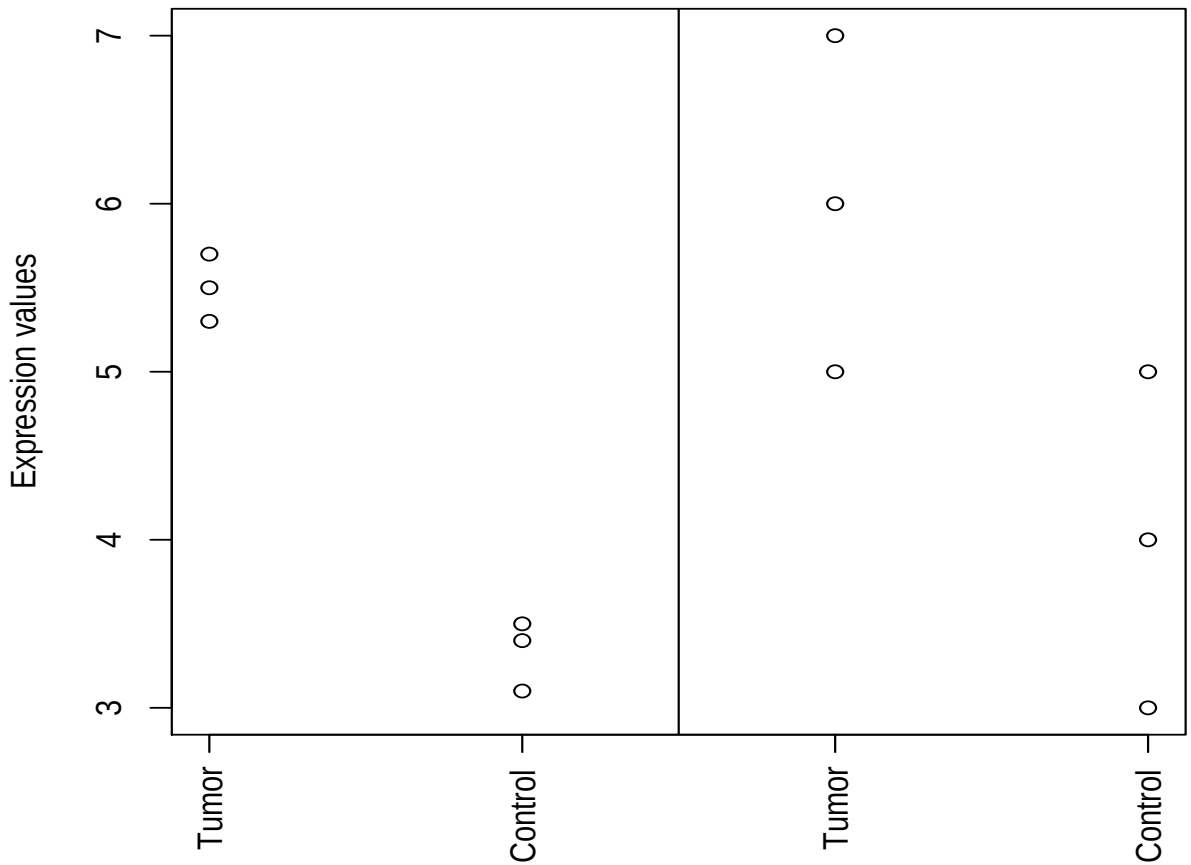
- Pros
  - Highly flexible
  - Increased power
    - Empirical Bayes
    - Linear modeling
    - Chip weighting
- Cons
  - Complexity
    - Design matrices
    - Contrast matrices

Simple Example

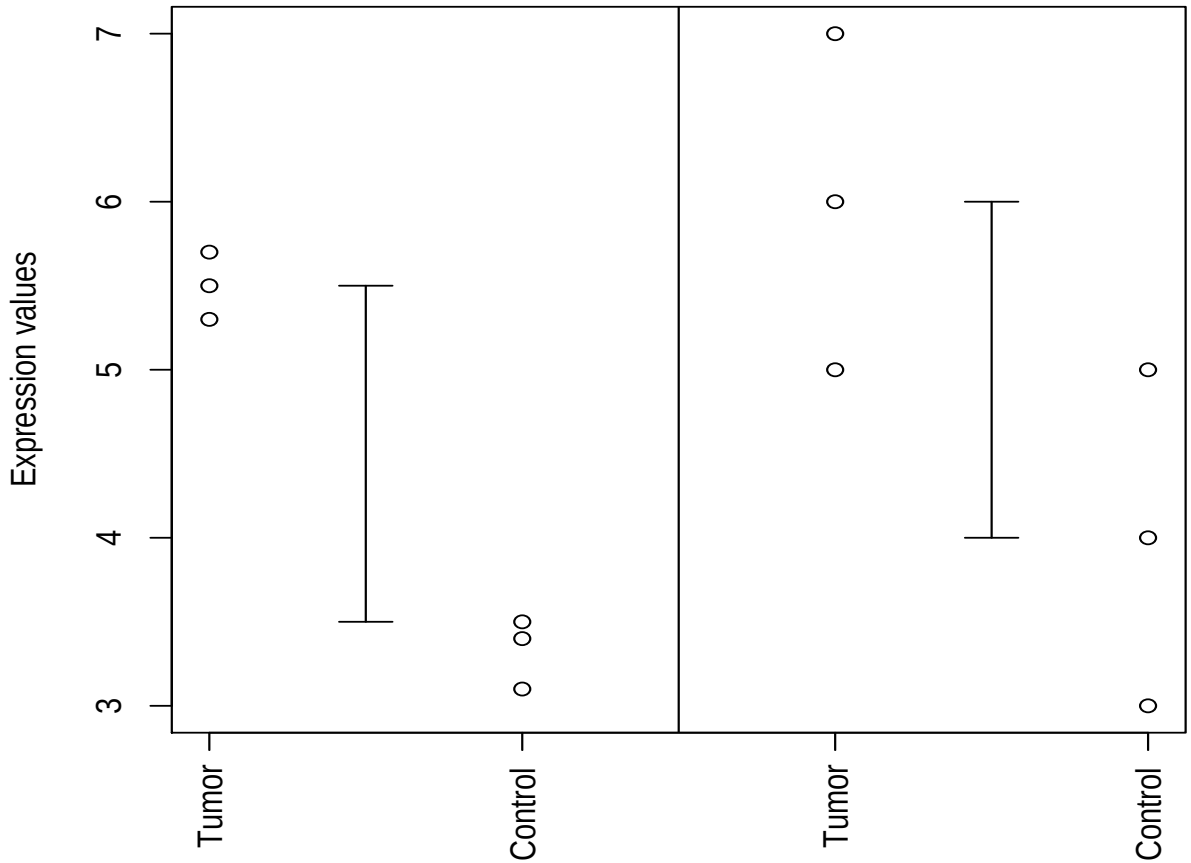
Compare two groups

$$t = \frac{\hat{x} - \hat{y}}{\frac{\hat{\sigma}}{\sqrt{N-1}}}$$

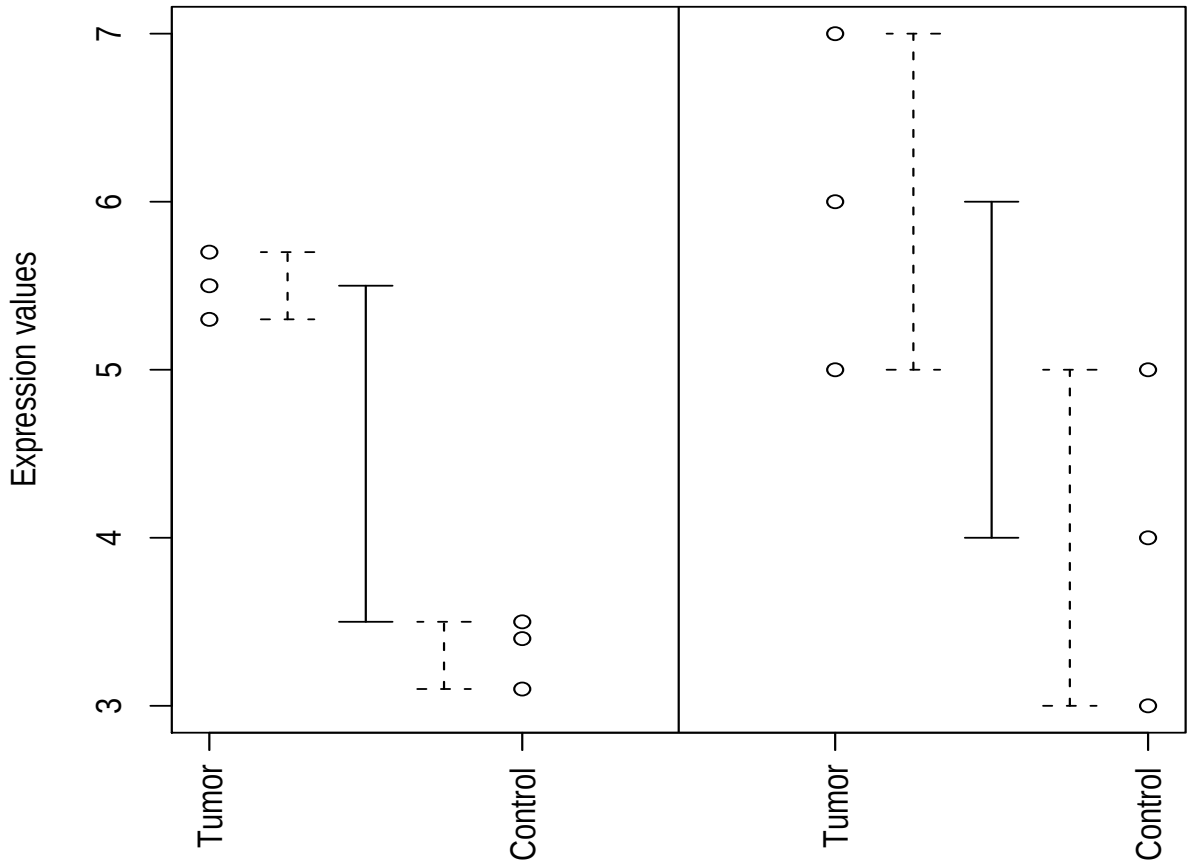
Graphical Example



Graphics



Graphics



Design Matrix


```
> samples
[1] Control Control Control Tumor   Tumor
[6] Tumor
Levels: Control Tumor
> design <- model.matrix(~0 + samples)
> colnames(design) <- levels(samples)
> design
  Control Tumor
1       1     0
2       1     0
3       1     0
4       0     1
5       0     1
6       0     1
attr(,"assign")
[1] 1 1
```

Contrast Matrix

Numerator

```
> contrast <- makeContrasts(Tumor - Control,
+                           levels = design)
> contrast
```

```
          Contrasts
Levels     Tumor - Control
  Control               -1
  Tumor                  1
```

Empirical Bayes

Denominator

Remember 'standard' t-test:

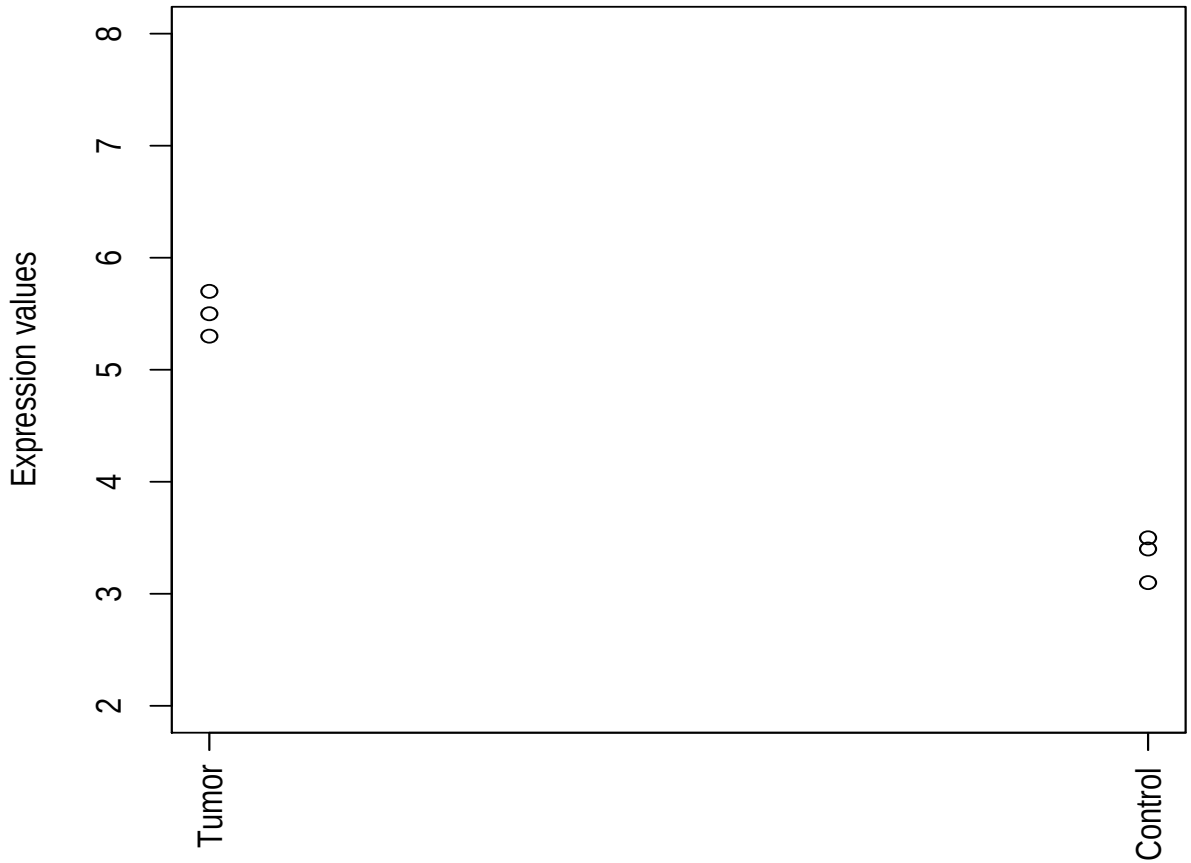
$$t = \frac{\hat{x} - \hat{y}}{\frac{\hat{\sigma}}{\sqrt{N-1}}}$$

limma uses Empirical Bayes adjusted denominator:

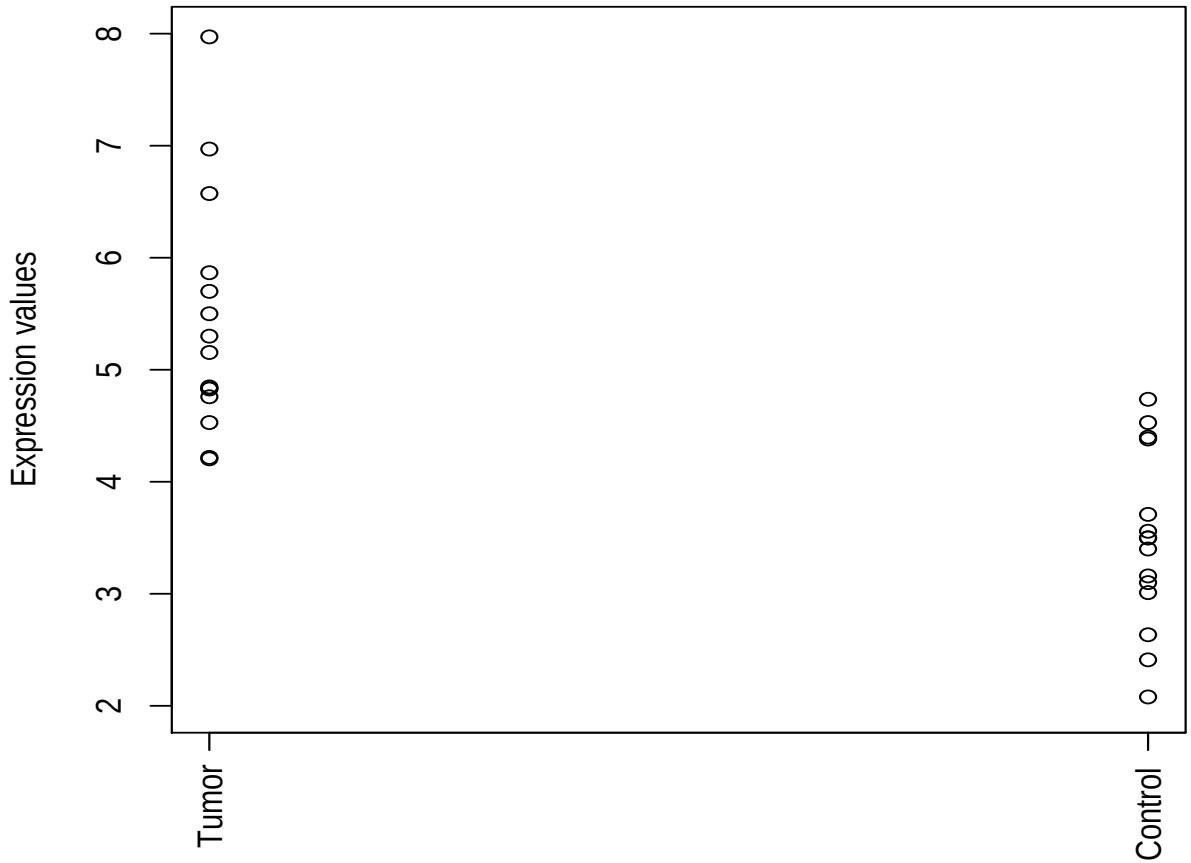
$$t = \frac{\hat{x} - \hat{y}}{s + s_0}$$

$$s = \frac{\hat{\sigma}}{\sqrt{N-1}}$$

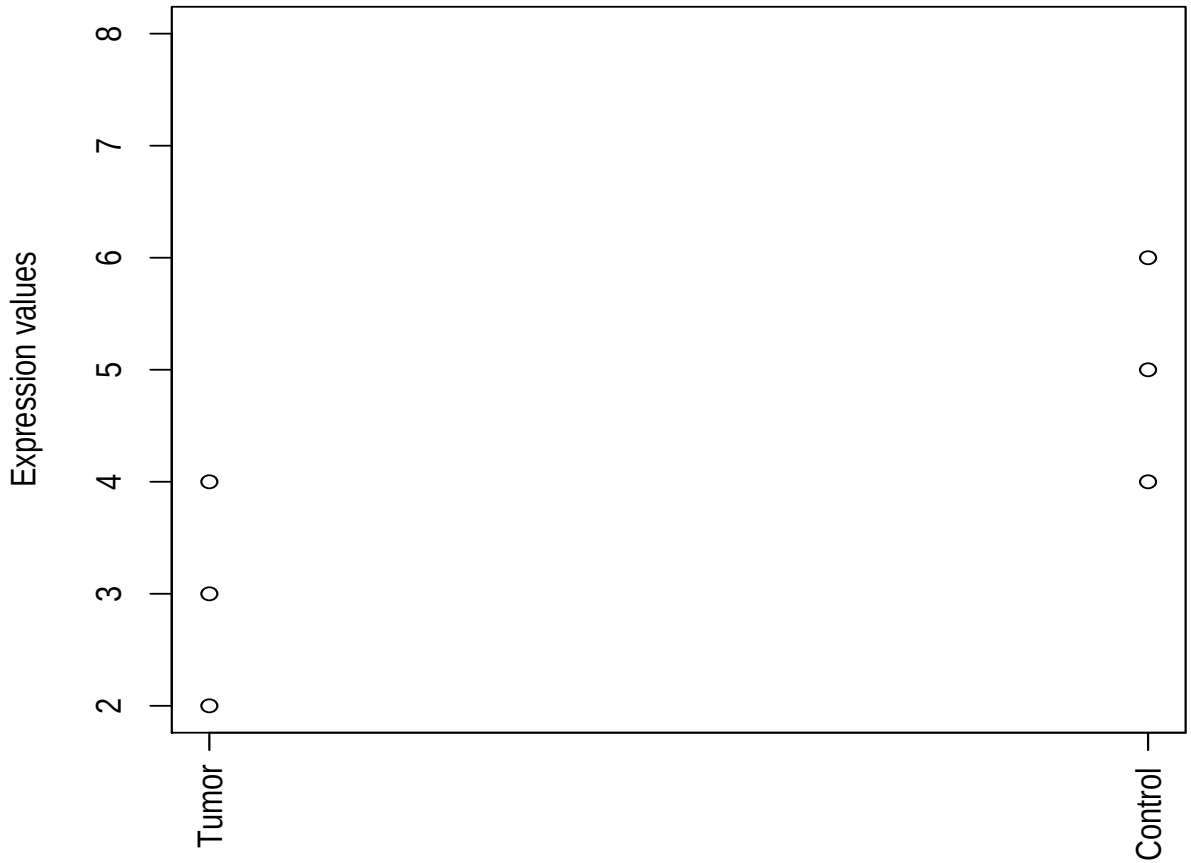
Why adjust?



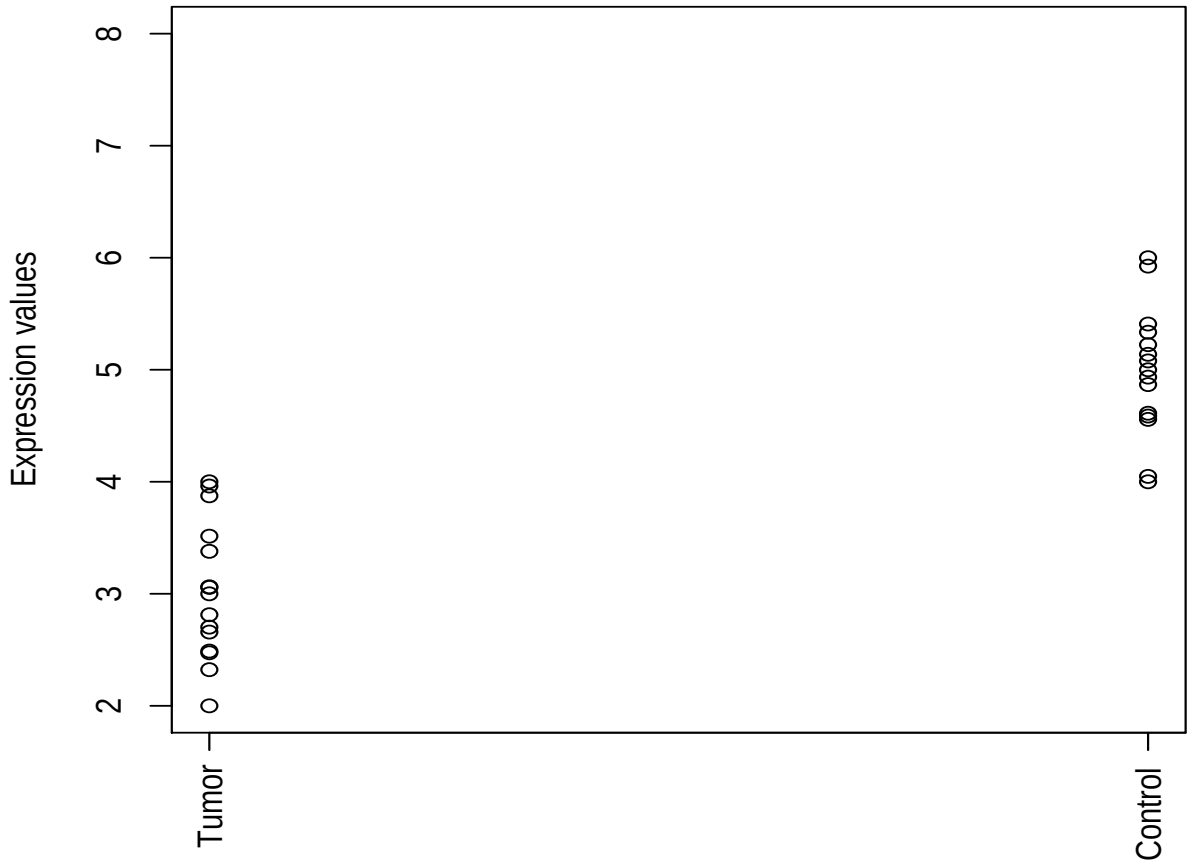
Why adjust?



Why adjust?



Why adjust?



t-test

Practice *t*-test

Load affy and limma libraries

Attach sample.ExpressionSet dataset

Look at phenoData object associated

Do a *t*-test comparing the male and female samples

t-test


```
> library(affy)
> library(limma)
> data(sample.ExpressionSet)
> eset <- sample.ExpressionSet
> head(pData(eset))
```

```
     sex    type score
A Female Control  0.75
B   Male    Case  0.40
C   Male Control  0.73
D   Male    Case  0.42
E Female    Case  0.93
F   Male Control  0.22
```

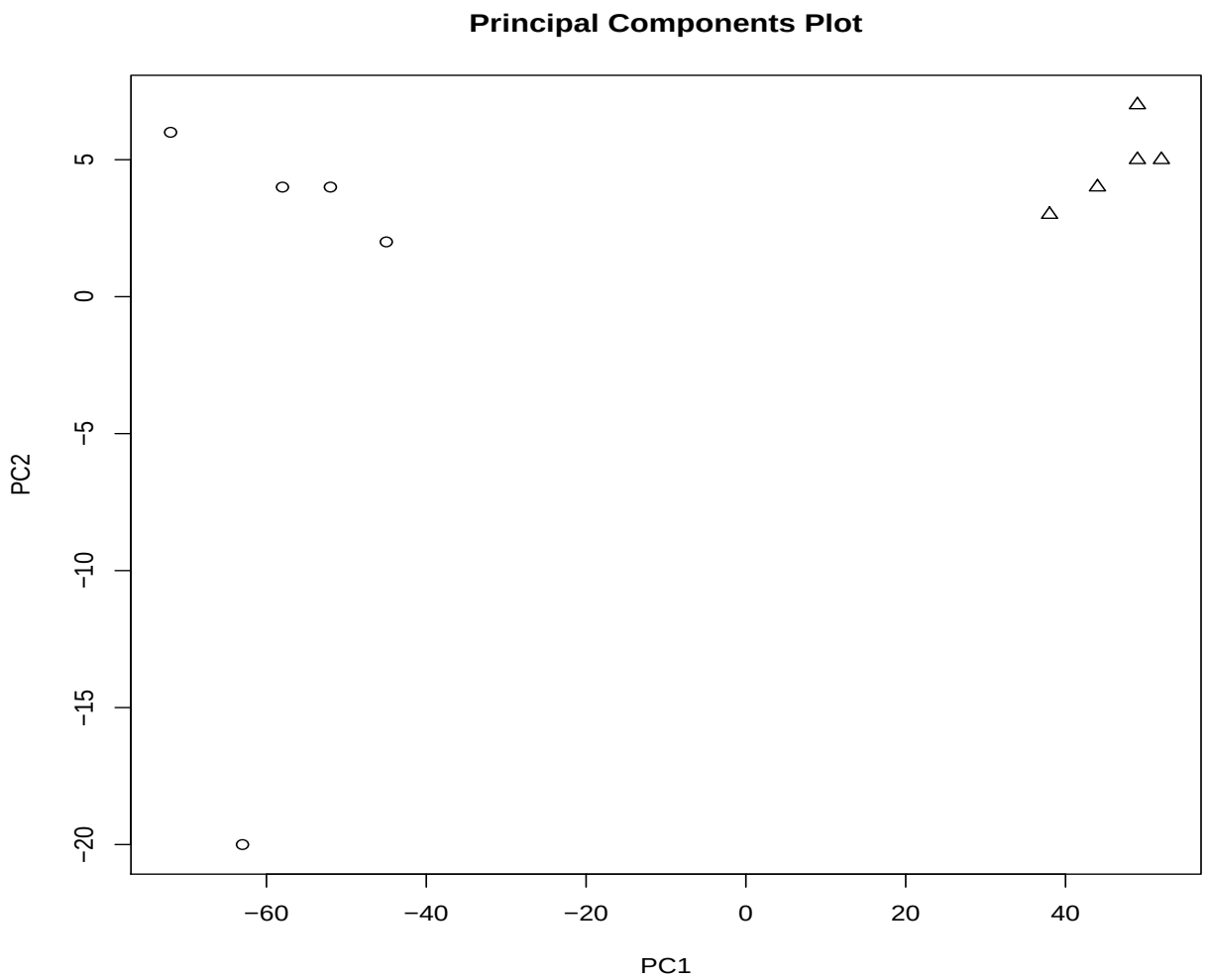
t-test


```
> design <- model.matrix(~ 0 + pData(eset)[,1])
> colnames(design) <- levels(pData(eset)[,1])
> contrast <- makeContrasts(Male - Female,
+                           levels = design)
> fit <- lmFit(eset, design)
> fit2 <- contrasts.fit(fit, contrast)
> fit2 <- eBayes(fit2)
```

t-test


```
> head(topTable(fit2, coef = 1))
                    ID logFC AveExpr    t
314         31553_at -11.3     7.2 -2.8
98          31337_at  97.2   422.8  2.6
303         31542_at   9.4    13.0  2.5
310         31549_at  -7.1    19.6 -2.2
4    AFFX-MurFAS_at  -7.3    13.9 -2.2
149         31388_at  16.4    33.2  2.2
    P.Value adj.P.Val    B
314   0.009      0.99 -4.6
98    0.016      0.99 -4.6
303   0.022      0.99 -4.6
310   0.036      0.99 -4.6
4     0.036      0.99 -4.6
149   0.040      0.99 -4.6
```

Dealing with Outliers



Array Weights


```
> aw <- arrayWeights(eset, design)
> aw
```

```
   1    2    3    4    5    6    7    8    9
0.81 0.69 1.06 1.46 1.68 1.37 0.65 0.83 0.72
  10   11   12   13   14   15   16   17   18
0.83 1.04 2.58 0.82 0.84 1.43 1.11 0.99 0.73
  19   20   21   22   23   24   25   26
0.63 0.76 0.84 0.68 2.65 1.14 1.25 0.73
```

```
> fit <- lmFit(eset, design, weights = aw)
```

Two-factor ANOVA

sample.ExpressionSet Again

```
> head(pData(eset))

     sex    type score
A Female Control  0.75
B   Male    Case  0.40
C   Male Control  0.73
D   Male    Case  0.42
E Female    Case  0.93
F   Male Control  0.22
```

Design Matrix


```
> sex <- pData(eset)[,1]
> type <- pData(eset)[,2]
> design <- model.matrix(~ 0 + sex:type)
> colnames(design) <- c("Fem.Case","Male.Case",
+                       "Fem.Contr","Male.Contr")
> head(design)
```

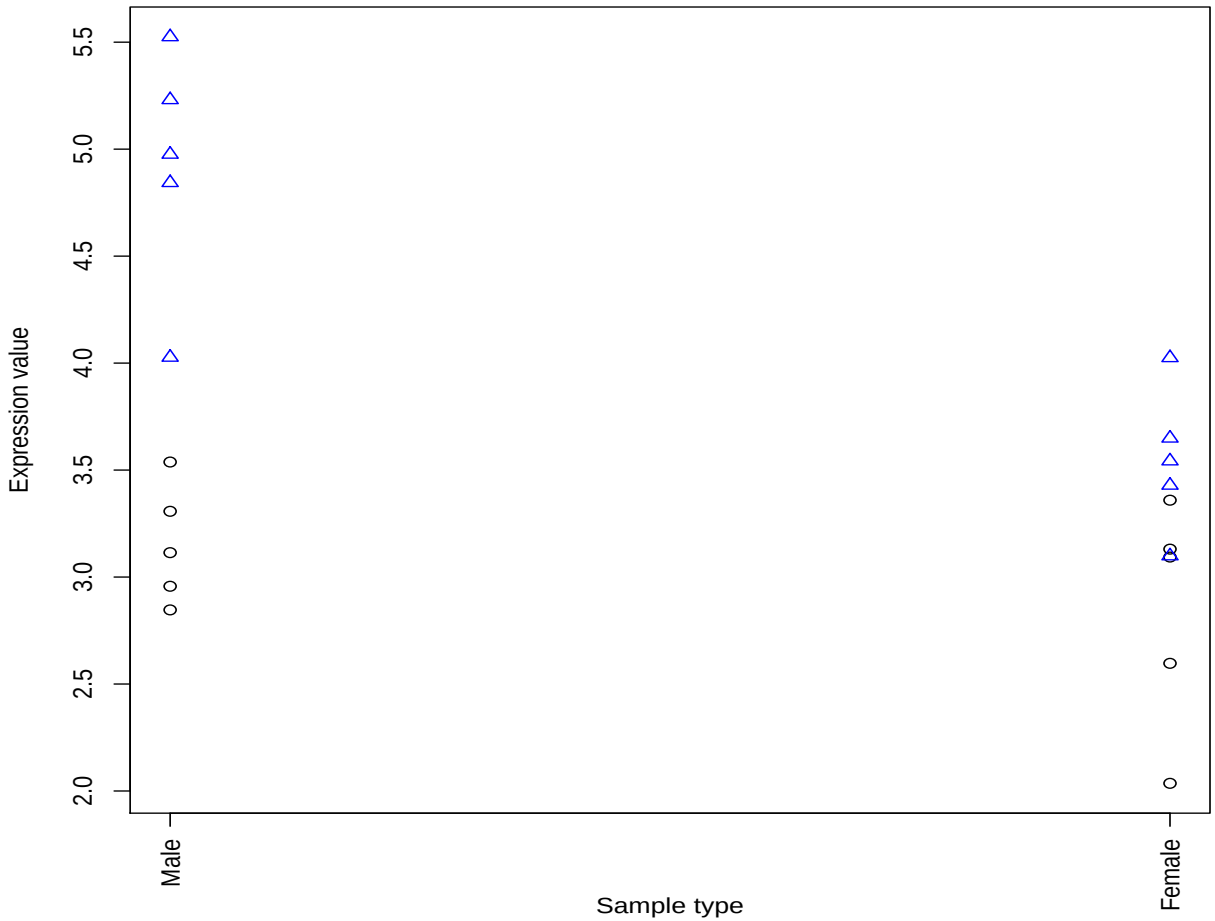
	Fem.Case	Male.Case	Fem.Contr	Male.Contr
1	0	0	1	0
2	0	1	0	0
3	0	0	0	1
4	0	1	0	0
5	1	0	0	0
6	0	0	0	1

Comparisons

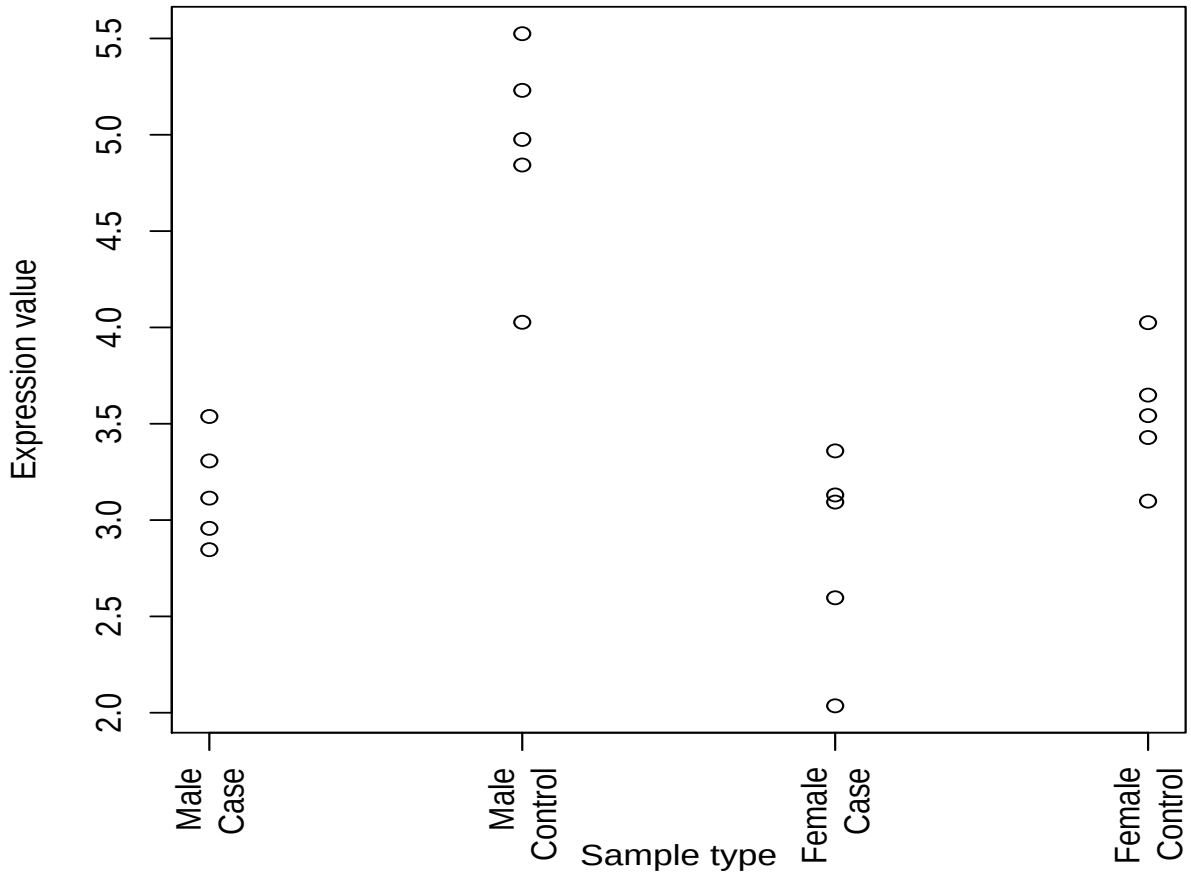
What comparisons can we make?

- Male vs Female
- Case vs Control
- Case vs Control within sex
- Interaction

t-test


Expression value
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
Male
Female
Sample type

ANOVA


Expression value
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
Male Case
Male Control
Female Case
Female Control
Sample type

Practice ANOVA

Can you do the following?

- Compare Female Cases vs Female Controls
- Compare Female Cases vs Male Cases

Create Contrasts Matrices


```
> contrast <- makeContrasts(Fem.Case - Fem.Contr,
+                           Fem.Case - Male.Case,
+                           levels = design)
> contrast
             Contrasts
Levels        Fem.Case - Fem.Contr
  Fem.Case                       1
  Male.Case                      0
  Fem.Contr                     -1
  Male.Contr                     0
             Contrasts
Levels        Fem.Case - Male.Case
  Fem.Case                       1
  Male.Case                     -1
  Fem.Contr                      0
  Male.Contr                     0
```

Fit Model and Compute Contrasts


```
> fit <- lmFit(eset, design)
> fit2 <- contrasts.fit(fit, contrast)
> fit2 <- eBayes(fit2)
```

Female Cases vs Controls

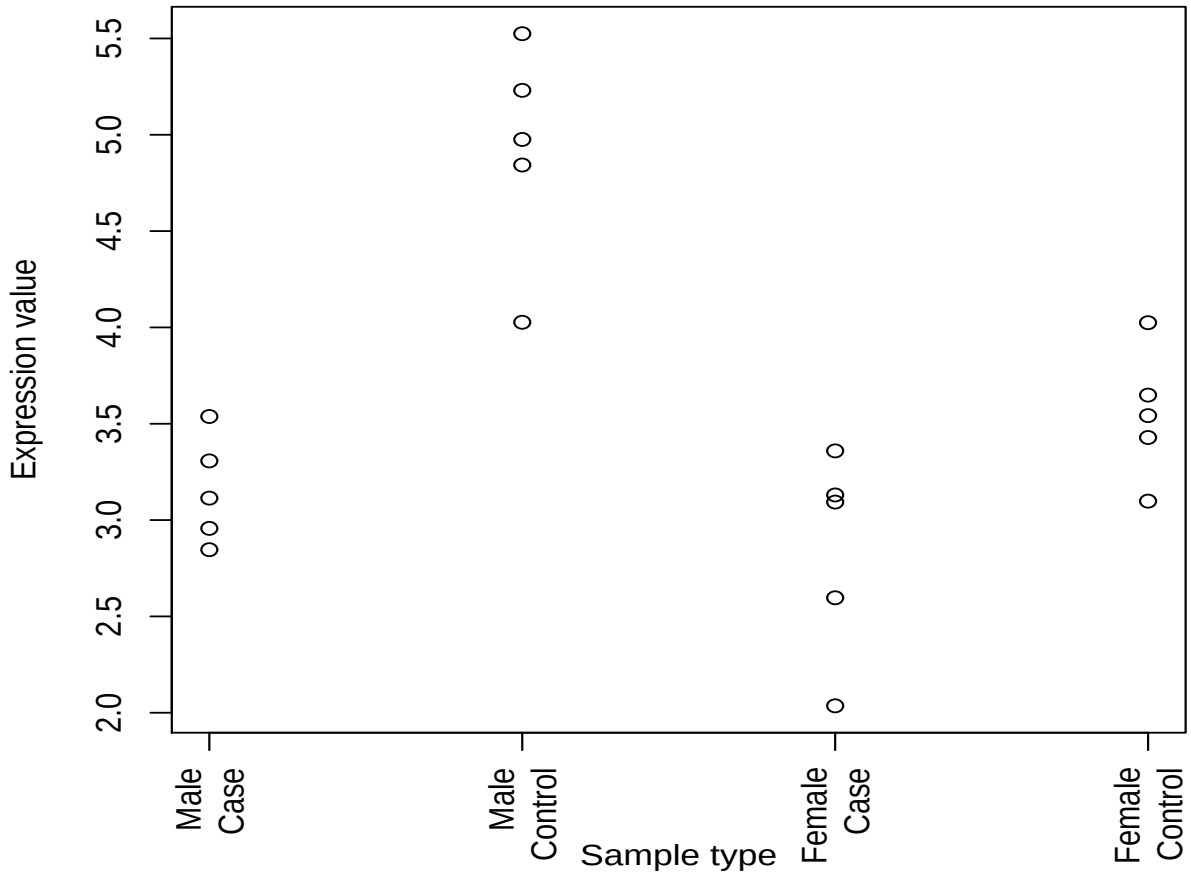

```
> head(topTable(fit2, coef = 1))
```

	ID	logFC	AveExpr	t	P.Value	adj.P.Val	B
180	31419_r_at	-395	1447	-2.8	0.010	0.75	-4.6
392	31631_f_at	27	-33	2.7	0.012	0.75	-4.6
113	31352_at	-27	40	-2.7	0.013	0.75	-4.6
374	31613_at	-44	77	-2.6	0.016	0.75	-4.6
358	31597_r_at	-335	1634	-2.5	0.022	0.75	-4.6
157	31396_r_at	-525	2504	-2.4	0.026	0.75	-4.6

Female Cases vs Male Cases


```
> head(topTable(fit2, coef = 2))
                                ID  logFC AveExpr
314                    31553_at   17.2     7.2
382                  31621_s_at -142.8   516.6
206                    31445_at  -63.5   134.8
120                    31359_at   11.7    11.9
43   AFFX-HUMRGE/M10098_5_at  -53.0    15.0
79                     31318_at   -9.7    12.7
       t P.Value adj.P.Val    B
314  3.6  0.0016      0.49 -4.4
382 -3.5  0.0020      0.49 -4.4
206 -2.9  0.0075      0.93 -4.5
120  2.5  0.0209      0.93 -4.5
43  -2.5  0.0217      0.93 -4.5
79  -2.4  0.0235      0.93 -4.5
```

Interaction



Interaction

$$interaction = (FemCase - FemContr) - (MaleCase - MaleContr)$$

Set up contrasts the same way:

```
> contrast <- makeContrasts((Fem.Case - Fem.Contr) -
+                           (Male.Case - Male.Contr),
+                           levels = design)
> colnames(contrast) <- "Interaction"
> contrast
```

```
            Contrasts
Levels       Interaction
  Fem.Case             1
  Male.Case           -1
  Fem.Contr           -1
  Male.Contr           1
```

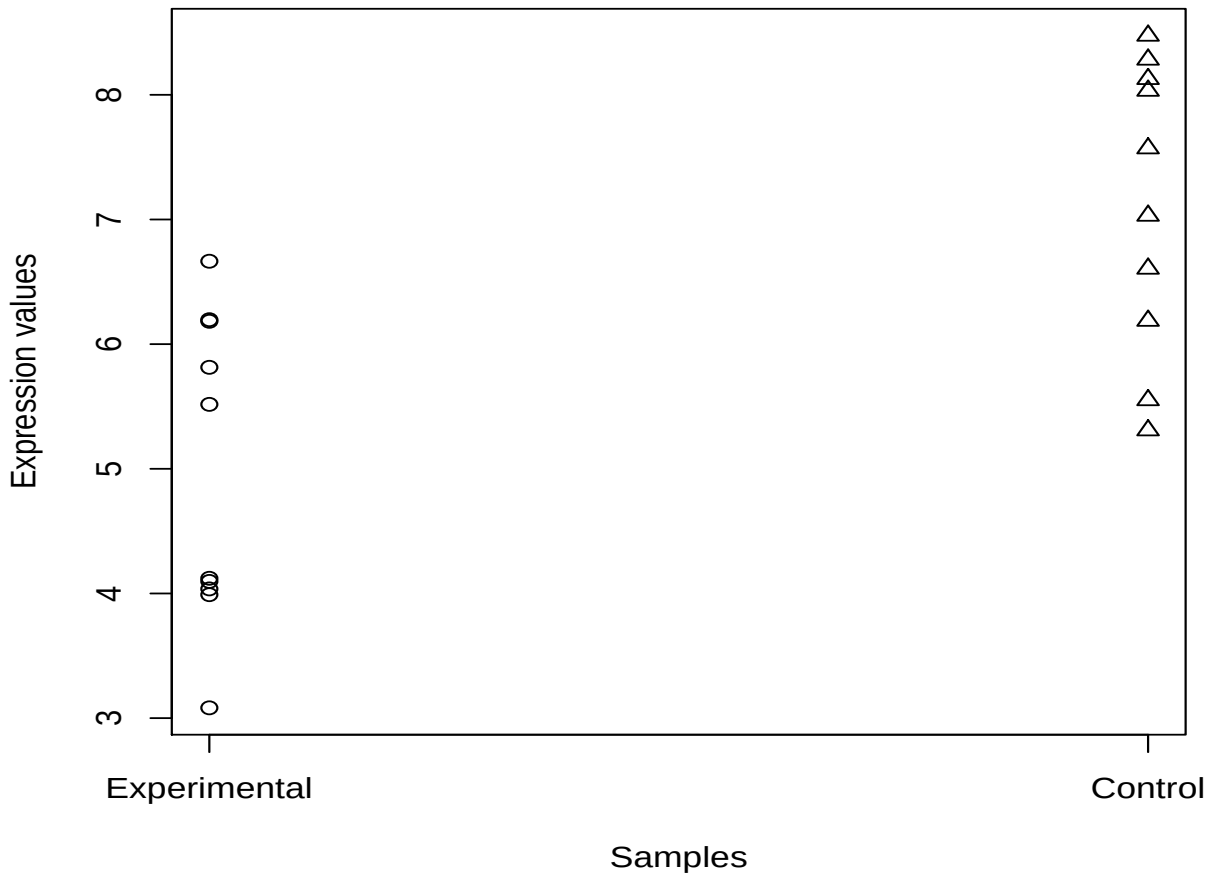
Batch Effects

- Batch effects can arise from
  - Pairing
  - Experiments run at different times
  - Different reagents
- Watch out for
  - Aliasing
  - Creating batches unnecessarily
  - Assuming batch effect when there isn't one

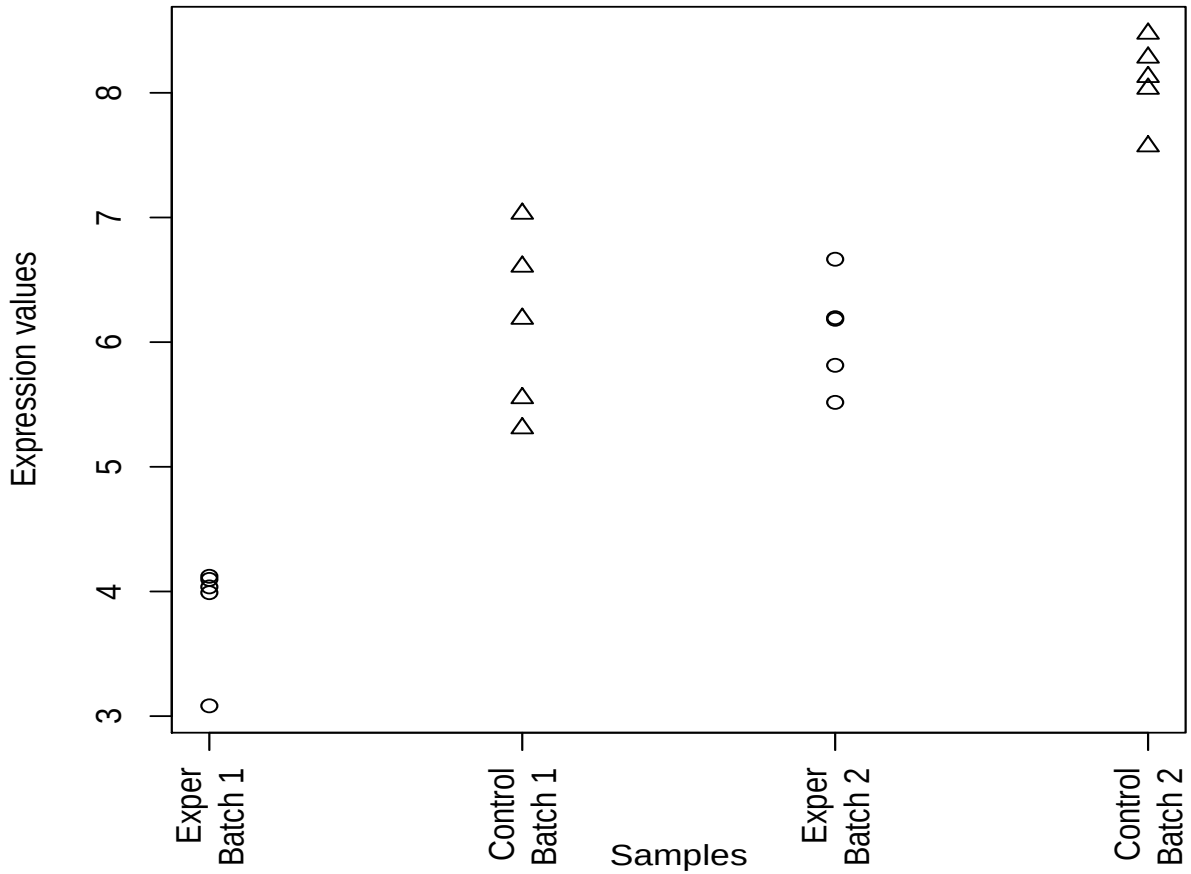
Pairing

- Mice first sampled as control
  then tumor introduced and re-sampled
- Wild type and mutant mice selected
  from several litters
- Several different cell lines treated similarly

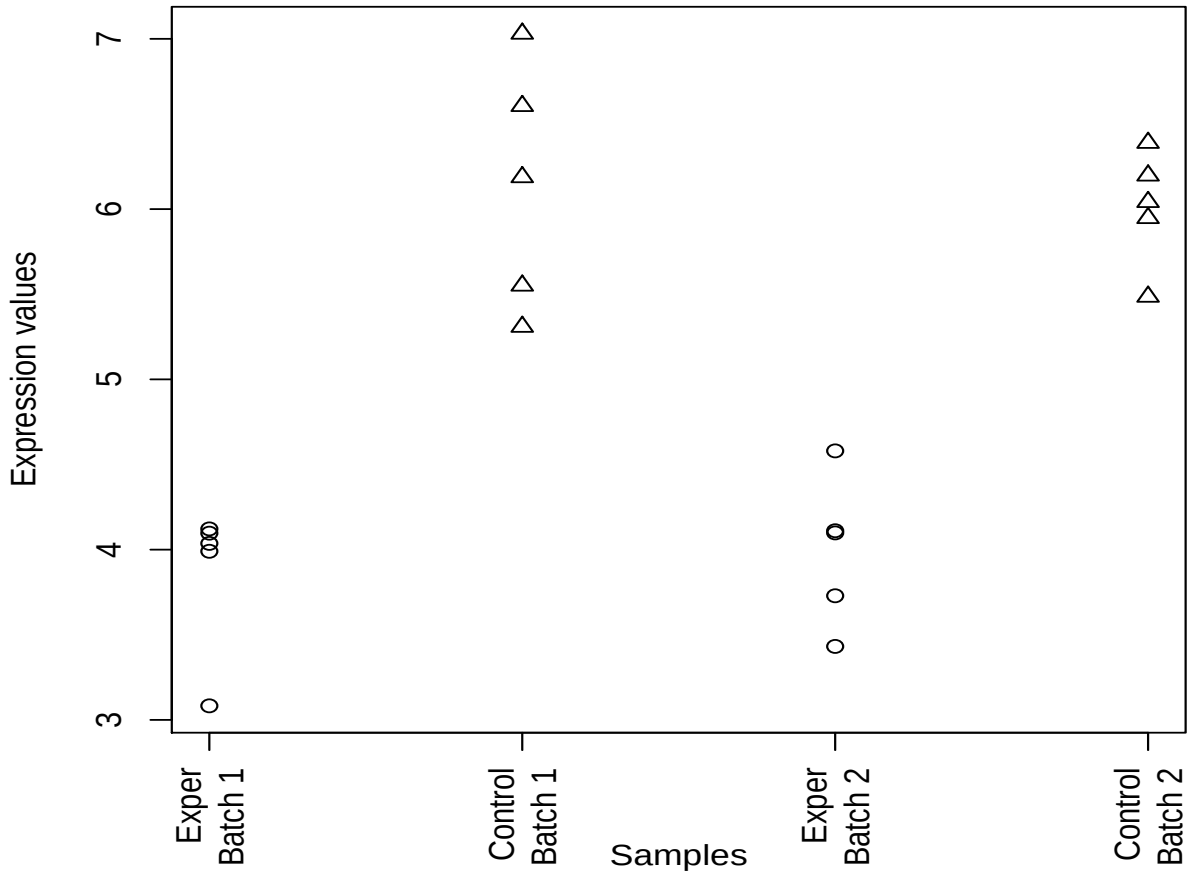
Example Batch Effect



Example Batch Effect



Controlling for Batch



Fitting a Batch Effect


```
> treatment <- factor(rep(1:2, each = 12))
> treatment
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
[21] 2 2 2 2
Levels: 1 2
> batch <- factor(rep(1:2, each = 6, times = 2))
> batch
 [1] 1 1 1 1 1 1 2 2 2 2 2 2 1 1 1 1 1 1 2 2
[21] 2 2 2 2
Levels: 1 2
```

Fitting a Batch Effect


```
> design <- model.matrix(~ 0 + treatment + batch)
> head(design)
  treatment1 treatment2 batch2
1          1          0      0
2          1          0      0
3          1          0      0
4          1          0      0
5          1          0      0
6          1          0      0
```

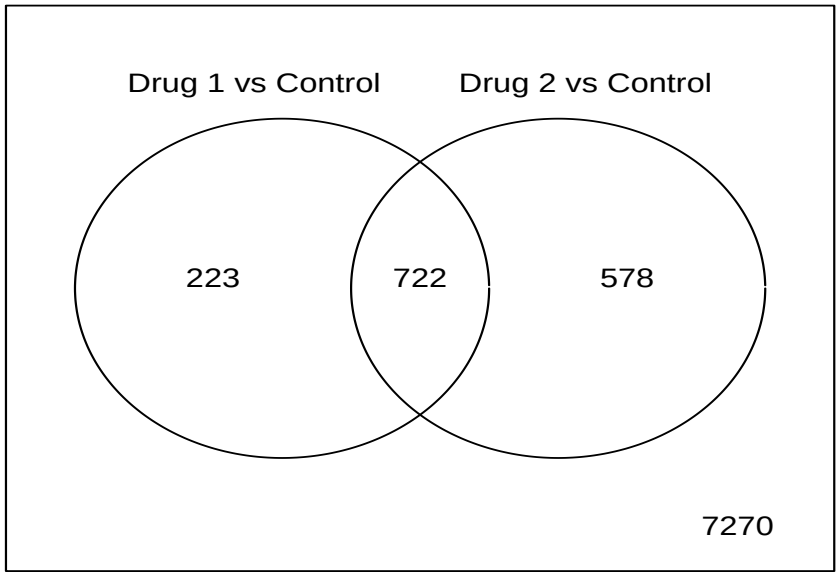
Multiple Comparisons

Setup:
We have compared two different drugs vs control and want to select significant genes

- decideTests()
  - separate
  - global
  - hierarchical
  - nestedF

Venn Diagram


```
> rslt <- decideTests(fit2, method = "nestedF")
> vc <- vennCounts(rslt)
> vennDiagram(vc)
```



Volcano Plot


```
> volcanoplot(fit2)
```