Using limma for Differential Expression

James W. MacDonald
jmacdon@med.umich.edu

BioC 2009
July 27, 2009

Overview

Typical analysis using limma:

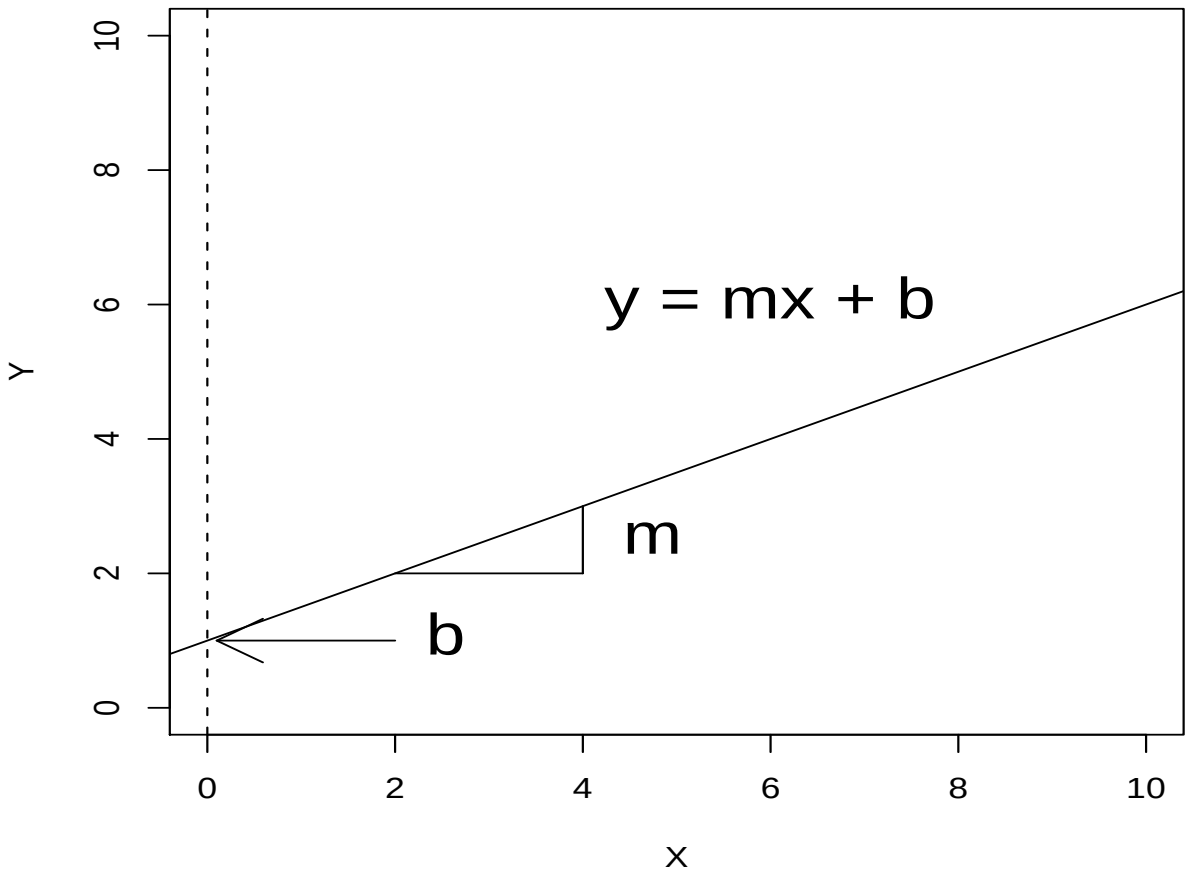
- Read in data
- Preprocess two-color data
- Create design matrix
- Create contrast matrix
- Fit model
- Make comparisons
- Output interesting results

Goals

Goals for this workshop:

- Statistics
  - Linear models (esp ANOVA)
  - Design matrices
  - Contrast matrices
- Other considerations
  - Intra-group correlation
  - Array weights

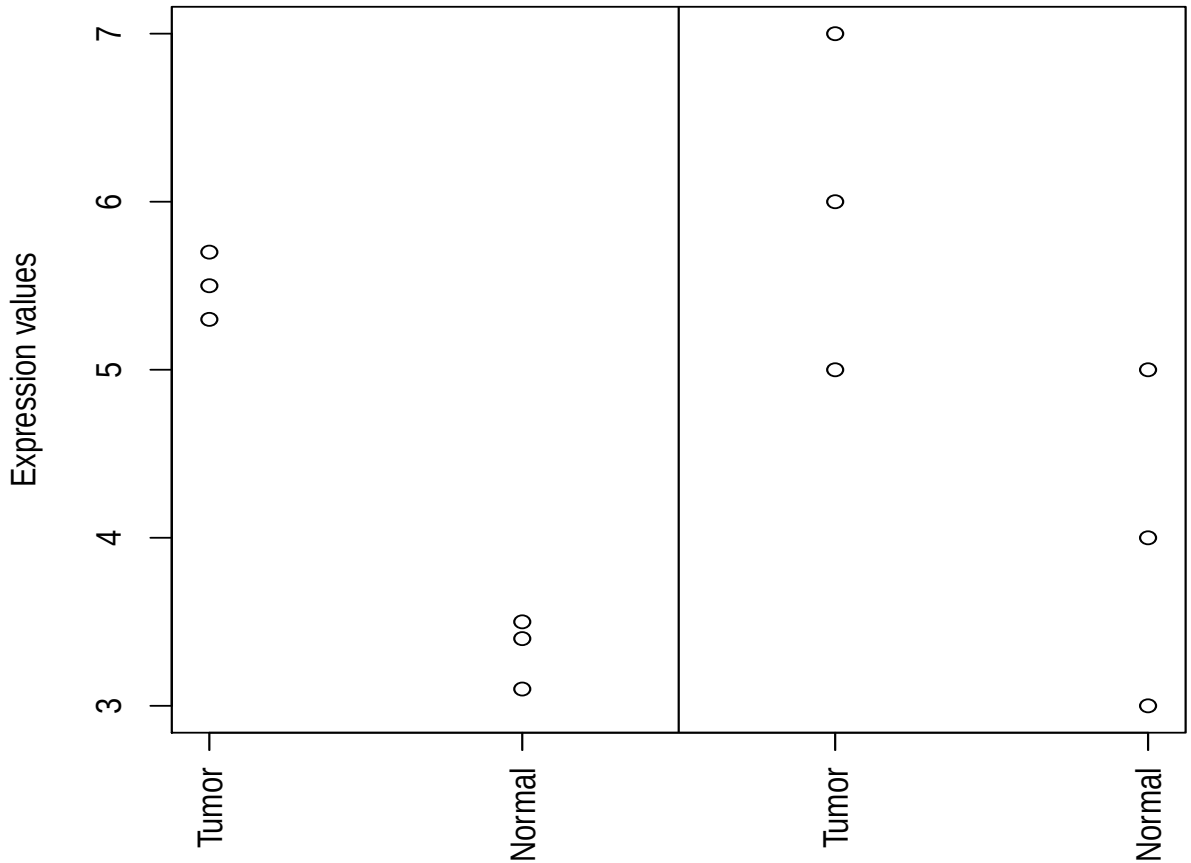
Linear model



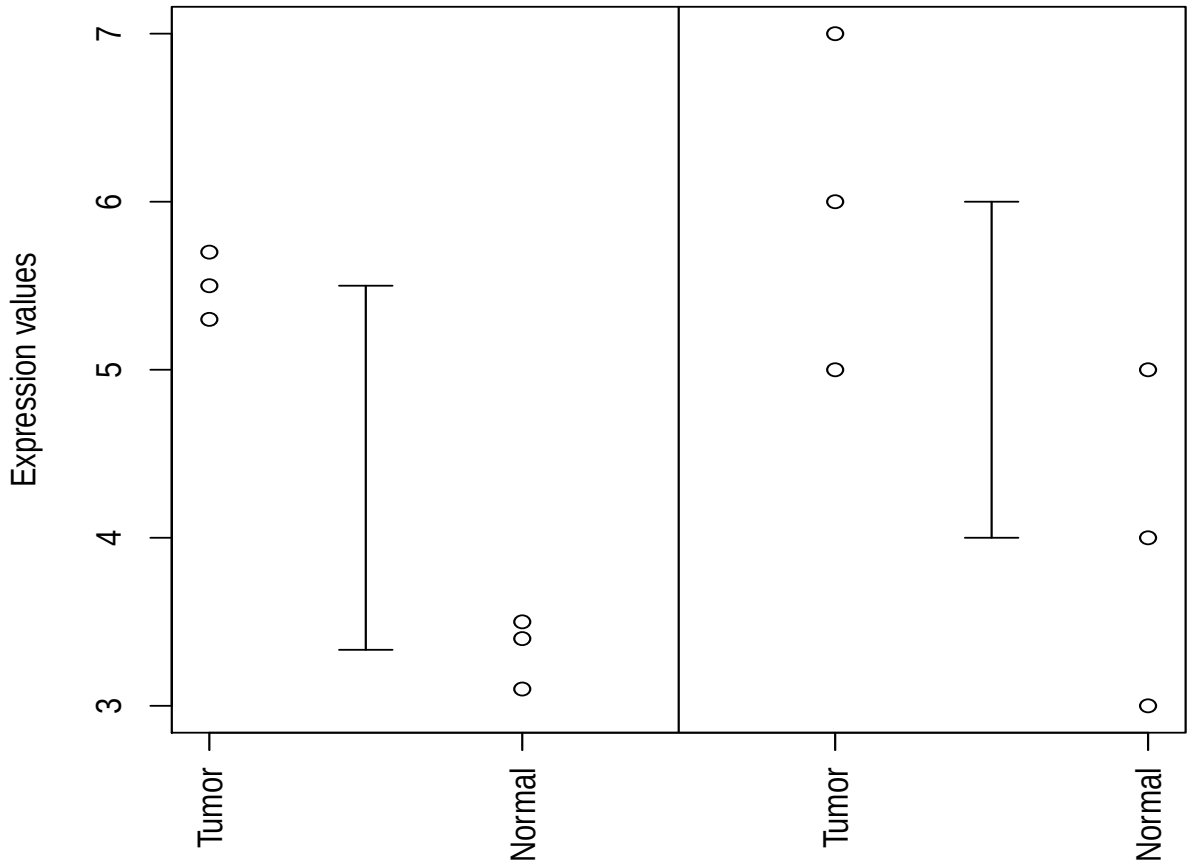
ANOVA

- Predictors are *factors* (unordered)
  - tumor/normal
  - experimental/control
  - mutant/wild type
- Simplest form of ANOVA is t-test

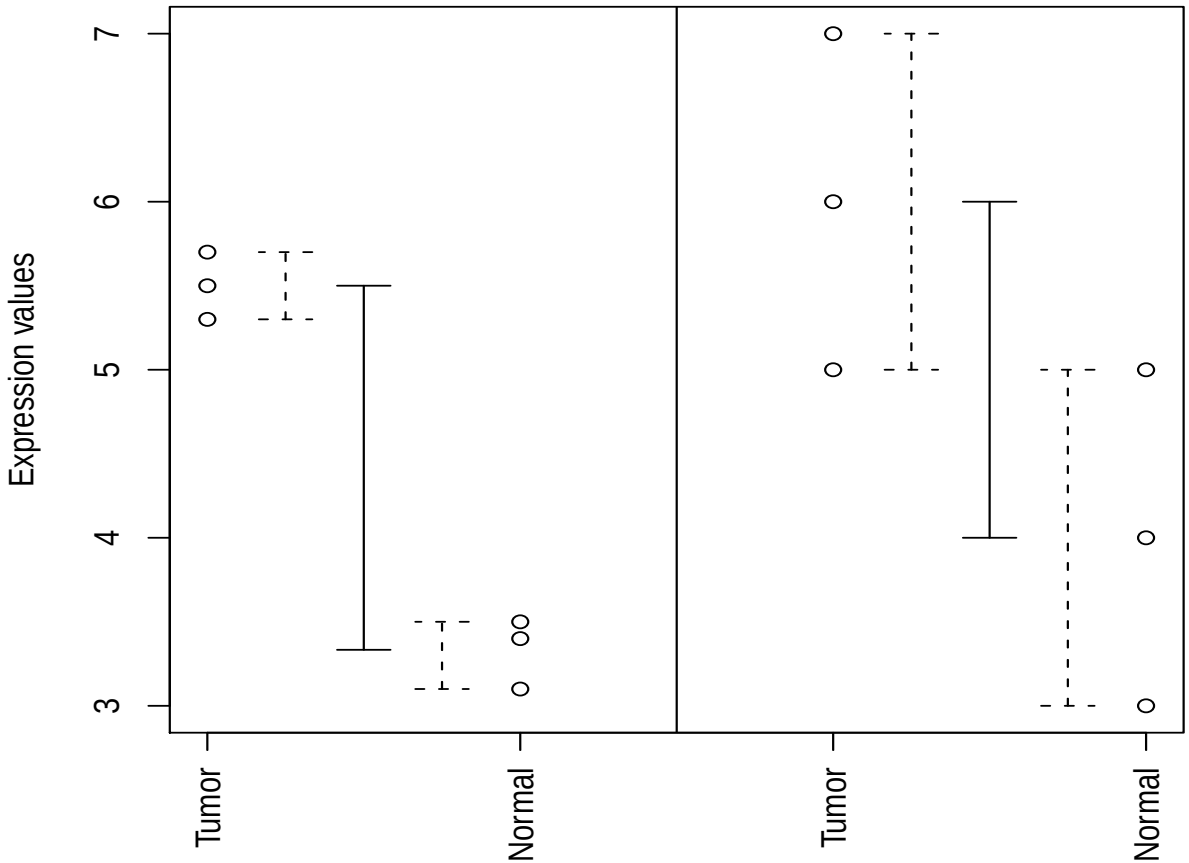
ANOVA



ANOVA



ANOVA



ANOVA

Basic form of the t-test:

$$\frac{Differencebetweengroupmeans}{Variabilitywithingroups} \tag{1}$$

ANOVA Model

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

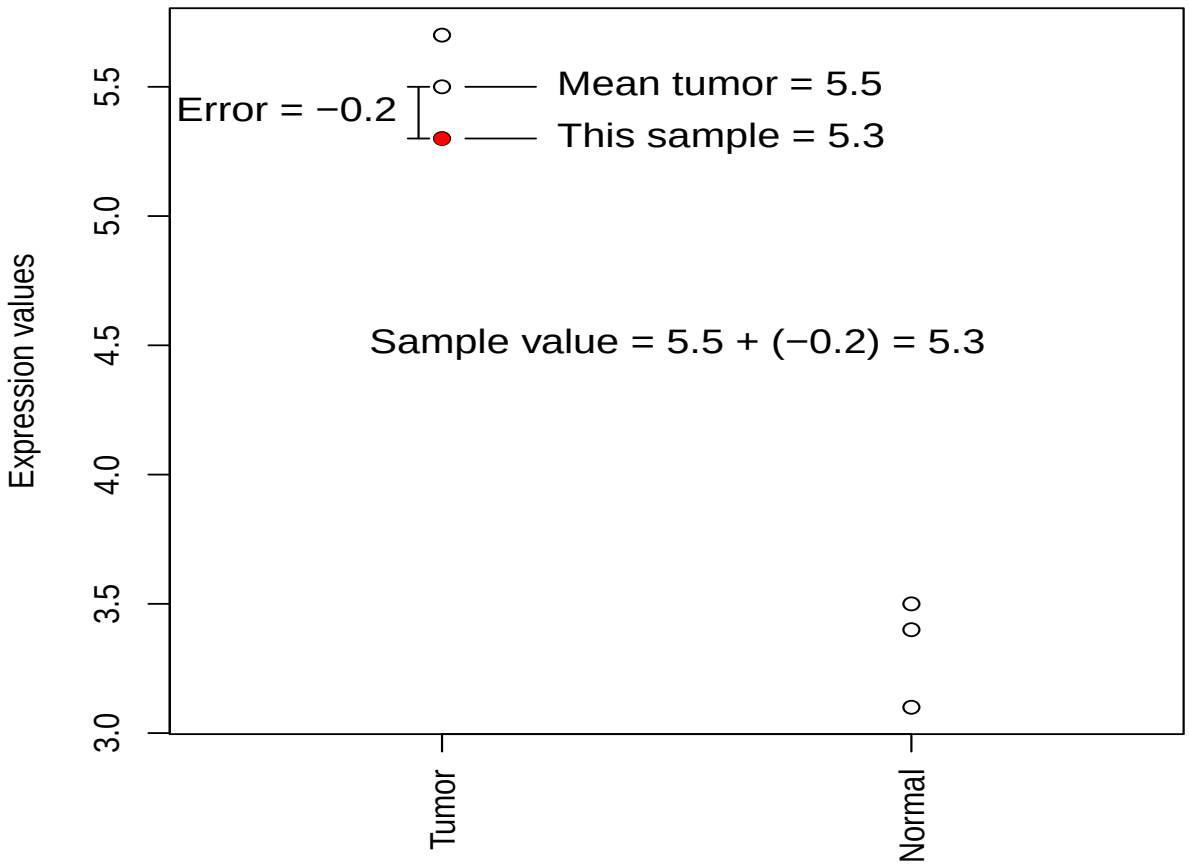
– OR –

tumor sample = mean tumor value + error

normal sample = mean normal value + error

ANOVA (Graphically)

ANOVA is a 'mean decomposition' of observed data



Linear Algebra - Background

Basic algebra – we need N equations to solve for N unknowns

$4 = y - 3x$

$1 = -2y + 4x$

Solve one equation for y, insert answer in other equation

$y = 3x - 4$

$1 = -2(3x + 4) + 4x$

$2x = -9$

$x = -9/2$

$y = -19/2$

We can do the same thing using matrix multiplication:

$$\begin{bmatrix} 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & -3 \\ -2 & 4 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix}$$

Linear Algebra

Solve for x and y:

$$\begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} 1 & -3 \\ -2 & 4 \end{bmatrix}^{-1} \begin{bmatrix} 4 \\ 1 \end{bmatrix}$$

In R:

```
> mat <- matrix(c(1,-2,-3,4), ncol = 2)
> solve(mat) %*% c(4,1)
```

```
     [,1]
[1,] -9.5
[2,] -4.5
```

ANOVA Model (again)

Our current model:

tumor sample = mean tumor value + error

normal sample = mean normal value + error

ANOVA Model (again)

Substitute x and y:

tumor sample $=$ x $+$ error

normal sample $=$ y $+$ error

ANOVA Model (again)

Expand a bit:

tumor sample $= 1x + 0y +$ error

normal sample $= 0x + 1y +$ error

ANOVA Model (again)

Plug in data:

$$5.3 = 1x + 0y + \text{error}$$
$$5.5 = 1x + 0y + \text{error}$$
$$5.7 = 1x + 0y + \text{error}$$
$$3.5 = 0x + 1y + \text{error}$$
$$3.1 = 0x + 1y + \text{error}$$
$$3.4 = 0x + 1y + \text{error}$$

ANOVA Model (again)

In linear algebra terms:

$$\begin{bmatrix} 5.3 \\ 5.5 \\ 5.7 \\ 3.5 \\ 3.1 \\ 3.4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} error \\ error \\ error \\ error \\ error \\ error \end{bmatrix}$$

First row is:

$5.3 = 1x + 0y + \text{error}$

ANOVA Model (again)

To check if our design matrix *really* does what we think...
Say we call our design matrix 'D'
And our vector of observations 'O'
To solve for (x,y) we use

$$\begin{bmatrix} x \\ y \end{bmatrix} = \left[\; D'D \; \right]^{-1} D'O$$

```
> mat <- matrix(rep(c(1,0,0,1), each = 3), ncol = 2)
> solve(crossprod(mat)) %*% t(mat)
```

```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 0.33 0.33 0.33 0.00 0.00 0.00
[2,] 0.00 0.00 0.00 0.33 0.33 0.33
```

Parameterization

- Parameter = thing estimated by model
  - Current model:
    - Mean of tumor samples
    - Mean of normal samples
  - Different model:
    - Mean of normal samples
    - Mean of tumor samples - mean of normal samples

Alternate Parameterization

Setup:

x = mean of tumor samples

y = mean of normal samples

z = mean of tumor - mean of normal

normal sample = 1y + 0z + error

tumor sample = 1y + 1z + error

so

tumor sample = 1y + (1x - 1y) + error

Alternate Parameterization

In linear algebra terms:

$$\begin{bmatrix} 5.3 \\ 5.5 \\ 5.7 \\ 3.5 \\ 3.1 \\ 3.4 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} + \begin{bmatrix} error \\ error \\ error \\ error \\ error \\ error \end{bmatrix}$$

Creating Design Matrices

By hand:

```
> mat <- cbind(c(1,1,1,0,0,0), c(0,0,0,1,1,1))
> dimnames(mat) <- list(paste("Sample", 1:6),
+                       c("Tumor","Normal"))
> mat
```

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

Creating Design Matrices

Using model.matrix():

```
> samps <- factor(rep(c("Tumor","Normal"), each = 3))
> model.matrix(~samps)
  (Intercept) sampsTumor
1           1          1
2           1          1
3           1          1
4           1          0
5           1          0
6           1          0
attr(,"assign")
[1] 0 1
attr(,"contrasts")
attr(,"contrasts")$samps
[1] "contr.treatment"
```

Creating Design Matrices

Using model.matrix():

```
> model.matrix(~0 + samps)

  sampsNormal sampsTumor
1           0          1
2           0          1
3           0          1
4           1          0
5           1          0
6           1          0
attr(,"assign")
[1] 1 1
attr(,"contrasts")
attr(,"contrasts")$samps
[1] "contr.treatment"
```

Creating Design Matrices

Decide on model, then create design matrix
Do not create design matrix and then figure out what the model is...

Contrast Matrices

What is a contrast?

- Difference between parameter estimates
- Coefficients must equal zero

Trivial example using our data:
1x - 1y

Contrast Matrices

Linear algebra again...

$$\begin{bmatrix} 5.34 & 2.91 \\ 6.54 & 8.56 \\ 2.35 & 2.56 \\ 5.55 & 6.45 \\ 8.78 & 6.98 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 2.43 \\ -2.02 \\ -0.21 \\ -0.9 \\ 1.8 \end{bmatrix}$$

Contrast Matrices

Rules:

- Coefficients must equal zero
- Rows of matrix must equal number of parameters in model

Contrast Matrices

Example: Three sample types (cont, trt1, trt2). Compare each trt to cont.

```
> contrast <- matrix(c(-1,1,0,-1,0,1), ncol = 2)
> dimnames(contrast) <- list(c("cont","trt1","trt2"),
+                            c("trt1 - cont",
+                              "trt2 - cont"))
> contrast
```

```
     trt1 - cont trt2 - cont
cont          -1          -1
trt1           1           0
trt2           0           1
```

Contrast Matrices

Example: Three sample types (cont, trt1, trt2). Compare the mean of trt samples to cont.

```
> contrast <- matrix(c(-1,0.5,0.5), ncol = 1)
> dimnames(contrast) <- list(c("cont","trt1","trt2"),
+                            "mean trt - cont")
> contrast
```

```
     mean trt - cont
cont            -1.0
trt1             0.5
trt2             0.5
```

Two Factor ANOVA

Setup:

- Two sample types (say, tumor and normal)
- Two treatments (say, chemo drug and 'carrier solution')
- Three replicates (12 samples total)
- Comparisons:
  - untreated tumor vs untreated normal
  - treated tumor vs untreated tumor
  - difference in effect of treatment

Two Factor ANOVA

Parameters we need:

- mean of treated normal samples
- mean of untreated normal samples
- mean of treated tumor samples
- mean of untreated tumor samples

Two Factor ANOVA

Design matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Two Factor ANOVA

In R:

```
> mat <- cbind(c(1,1,1,rep(0,9)),
+              c(0,0,0,1,1,1,rep(0,6)),
+              c(rep(0,6),1,1,1,0,0,0),
+              c(rep(0,9),1,1,1))
> colnames(mat) <- c("nu","nt","tu","tt")
```

Two Factor ANOVA


```
> mat
```

	nu	nt	tu	tt
[1,]	1	0	0	0
[2,]	1	0	0	0
[3,]	1	0	0	0
[4,]	0	1	0	0
[5,]	0	1	0	0
[6,]	0	1	0	0
[7,]	0	0	1	0
[8,]	0	0	1	0
[9,]	0	0	1	0
[10,]	0	0	0	1
[11,]	0	0	0	1
[12,]	0	0	0	1

Two Factor ANOVA

Contrast matrix:

```
> makeContrasts(tu - nu, tt - tu,
+                 tt - tu - nt + nu,
+                 levels = mat)
```

```
      Contrasts
Levels tu - nu tt - tu tt - tu - nt + nu
    nu      -1       0                 1
    nt       0       0                -1
    tu       1      -1                -1
    tt       0       1                 1
```

Two Factor ANOVA

Using model.matrix()

```
> typ <- factor(rep(c("Norm","Tum"),  each = 6))
> trt <- factor(rep(c("Untrt", "Trt"),each = 3, times
> typ
 [1] Norm Norm Norm Norm Norm Norm Tum  Tum
 [9] Tum  Tum  Tum  Tum
Levels: Norm Tum

> trt
 [1] Untrt Untrt Untrt Trt   Trt   Trt
 [7] Untrt Untrt Untrt Trt   Trt   Trt
Levels: Trt Untrt

> mat <- model.matrix(~0+trt*typ)
```

Two Factor ANOVA


```
> mat
   trtTrt trtUntrt typTum trtUntrt:typTum
1       0        1      0                0
2       0        1      0                0
3       0        1      0                0
4       1        0      0                0
5       1        0      0                0
6       1        0      0                0
7       0        1      1                1
8       0        1      1                1
9       0        1      1                1
10      1        0      1                0
11      1        0      1                0
12      1        0      1                0
attr(,"assign")
[1] 1 1 2 3
```

Two Factor ANOVA

Questions:

- What parameters are being estimated?
- What contrasts matrix do we need?

Batch Effect

Setup:

- Two sample types (say, treated and control)
- Two experiments, one week apart
- Three replicates (12 samples total)
- Compare treated vs normal

Batch Effects


```
> typ <- factor(rep(c("trt","cont"),
+                   each = 3, times = 2))
> bat <- factor(rep(1:2, each = 6))
> mat <- model.matrix(~ 0 + typ + bat)
```

Batch Effects


```
> mat
   typcont typtrt bat2
1        0      1    0
2        0      1    0
3        0      1    0
4        1      0    0
5        1      0    0
6        1      0    0
7        0      1    1
8        0      1    1
9        0      1    1
10       1      0    1
11       1      0    1
12       1      0    1
attr(,"assign")
[1] 1 1 2
```

Two-color Data

Design matrices are a bit more tricky:

- We are working with ratios
  - Max number of parameters = sample types - 1
- Dye swaps
- Experimental design considerations
- modelMatrix()

Two Samples

Setup:

- Two sample types (say, wild type and mutant)
- All mutant labeled with Cy5
- All wild type labeled with Cy3
- Four of each sample (four arrays)

Two Samples


```
> targets <- data.frame(Cy5 = rep("mut", 4),
+                       Cy3 = rep("wt", 4))
> modelMatrix(targets, ref = "wt")

Found unique target names:
 mut wt
     mut
[1,]   1
[2,]   1
[3,]   1
[4,]   1
```

Dye Swaps

Same setup, but every other array we swap dyes

Dye Swaps


```
> targets <- data.frame(Cy5 = rep(c("mut","wt"), 2),
+                       Cy3 = rep(c("wt","mut"),2))
> modelMatrix(targets, ref = "wt")

Found unique target names:
 mut wt
     mut
[1,]   1
[2,]  -1
[3,]   1
[4,]  -1
```

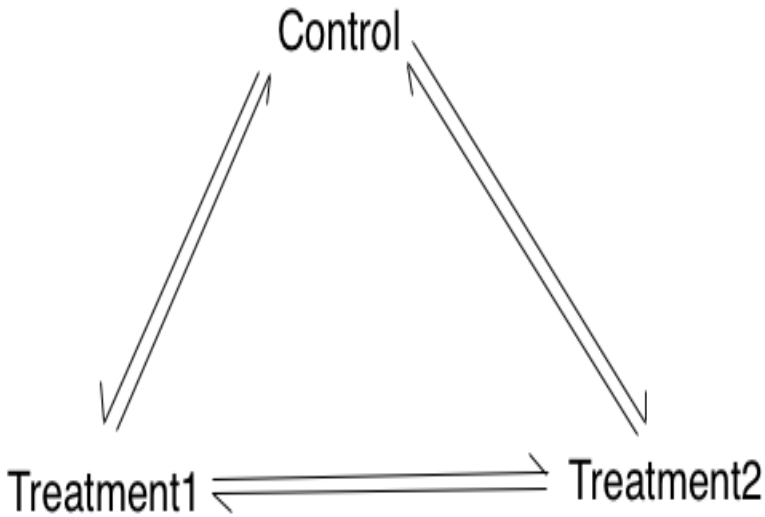
Dye Effect

Same setup, but we want to test for dye effect

```
> cbind(dye = 1, mut = rep(c(1,-1), 2))
```

```
     dye mut
[1,]   1   1
[2,]   1  -1
[3,]   1   1
[4,]   1  -1
```

'Connected' Experiment



'Connected' Experiment

Targets file:

```
   Cy3  Cy5
1 cont trt1
2 trt1 cont
3 trt1 trt2
4 trt2 trt1
5 trt2 cont
6 cont trt2
```

'Connected' Experiment

Design matrix:

```
> modelMatrix(tgts, ref = "cont")
```

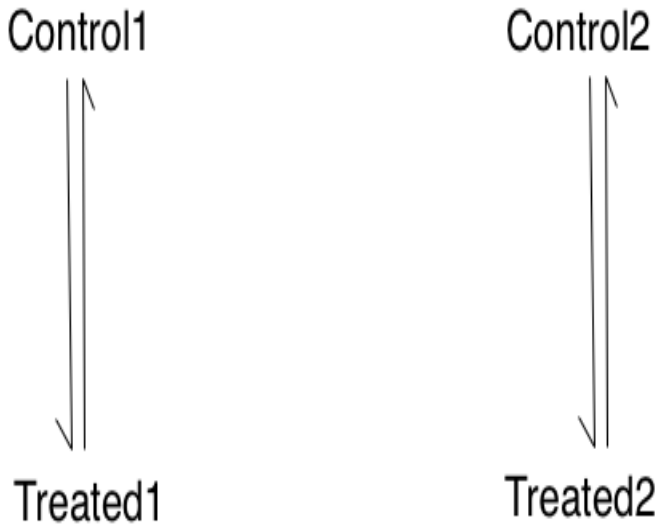
```
Found unique target names:
 cont trt1 trt2
     trt1 trt2
[1,]    1    0
[2,]   -1    0
[3,]   -1    1
[4,]    1   -1
[5,]    0   -1
[6,]    0    1
```

'Connected' Experiment

By Hand:

$$\begin{bmatrix} \textbf{Control} & \textbf{Trt1} & \textbf{Trt2} \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} \textbf{Control} & \textbf{Trt1} & \textbf{Trt2} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

'Unconnected' Experiment



'Unconnected' Experiment

Targets file:

```
    Cy5   Cy3
1 cont1  trt1
2  trt1 cont1
3 cont2  trt2
4  trt2 cont2
```

'Unconnected' Experiment

Incorrect design matrix:

```
> modelMatrix(tgts, ref = "cont1")

Found unique target names:
 cont1 cont2 trt1 trt2
     cont2 trt1 trt2
[1,]     0   -1    0
[2,]     0    1    0
[3,]     1    0   -1
[4,]    -1    0    1
```

'Unconnected' Experiment


```
> dat <- matrix(rnorm(40), ncol=4)
> fit <- lmFit(dat, modelMatrix(tgts, ref = "cont1"))
Found unique target names:
 cont1 cont2 trt1 trt2
Coefficients not estimable: trt2
> head(fit$coef)
     cont2  trt1 trt2
[1,] -1.91  0.27   NA
[2,] -0.27 -1.46   NA
[3,] -0.48  0.71   NA
[4,]  0.86 -0.22   NA
[5,] -1.11 -0.75   NA
[6,]  0.15  0.17   NA
```

Intra-group correlation

- Batch effect
- Technical replication
- Correlated samples
  - Multiple litters
  - Longitudinal data

Intra-group correlation

- Fixed batch effect
- Mixed model
  - Estimate correlation (duplicateCorrelation)
  - Fit model ('correlation' argument)

Array weights

- With arrays of lesser quality you can
  - Discard array data
  - Down-weight arrays
- Weights based on linear fit
- arrayWeights/arrayWeightsSimple
- 'weight' argument to lmFit()