GS01 0163
Analysis of Microarray Data

Keith Baggerly and Kevin Coombes
Section of Bioinformatics
Department of Biostatistics and Applied Mathematics
UT M. D. Anderson Cancer Center
kabagg@mdanderson.org
kcoombes@mdanderson.org

11 October 2005

Lecture 11: Differential Expression and Modelling

- Testing Redux, and One More
- Comparing three or more groups
- Pairing
- Incorporating covariates
- Models

A Rehash

Comparing two groups:

- t-tests, Wilcoxon tests

Correcting for multiple testing:

- permutation tests
- Bonferroni, BUM and Empirical Bayes

Changing the question:

- Tail Ranks and Biomarkers

One More Difference Measure...

Still looking at one gene, and two groups of measurements for that gene

t-tests let us say “these are different”, but do not necessarily let us say anything about “how different are they?”

We can form confidence intervals corresponding for a given difference (eg, diff in log ratios) and convert that confidence interval into another interval on a scale that is more meaningful to us (such as fold change).

Combining CIs and Criteria

Now, there's a neat trick that can be used here by combining confidence intervals with the quantity of interest.

Our question till now has been "is this gene differentially expressed between the two groups?", but we can expand this to include another criterion by asking "is this gene differentially expressed between the two groups by at least a minimal amount k?"

The dChip Approach

For each group, assemble point estimates of the expression levels. These point estimates are assumed to have normal distributions. We can then form a confidence interval for the ratio, and we can focus our attention just on those genes where the *lower bound* of this confidence interval is more than k-fold. Thus, not only are we pretty sure that the gene is differentially expressed, but we believe that it is different by at least a minimal amount that we can specify.

Is this the way to go?

I don't necessarily think the dChip answers are right, because I think that their model has the wrong error structure, but I do think that the confidence interval idea has some merit.

It has the practical advantage of using more than one filtering criterion to assess "significance".

Applying Bonferroni requires setting a very wide confidence interval. Permutation tests still work.

Expanding our Focus

Say we have data from 3 groups that were run at the same time, as opposed to 2. Does this change the outcome of our initial comparison of two groups?

- Given microarray experiments on
  - N_A sample of type A
  - N_B sample of type B
  - N_C sample of type C
- Decide which of the G genes on the microarray are differentially expressed between groups A and B.

Expanding our Focus

The t-statistic from before

$$t = \frac{\bar{x}_B - \bar{x}_A}{s_P \sqrt{1/N_A + 1/N_B}}.$$

The numerator doesn't change, but what about the denominator?

The pooled estimate of the standard deviation initially includes data from just A and B, but it can be expanded to include data from all of the groups

The Broader Pool...

For two groups:

$$s_P^2 = \frac{(N_A - 1)s_A^2 + (N_B - 1)s_B^2}{N_A + N_B - 2}.$$

For three groups:

$$s_P^2 = \frac{(N_A - 1)s_A^2 + (N_B - 1)s_B^2 + (N_C - 1)s_C^2}{N_A + N_B + N_C - 3}.$$

What Does This Buy Us?

A more precise estimate of the variation gives us more degrees of freedom for the t-test.

More degress of freedom gives us a more sensitive test.

Extreme case: $N_A = 2, N_B = 2, N_C = 10$.

How many differences do we see?

Some Simulations

No differences in the data...

```
n.genes <- 2000
an <- 2; bn <- 2; cn <- 10
n.samples <- an + bn + cn;
type <- factor(rep(c('A', 'B', 'C'),
                   times=c(an, bn, cn)))
data <- matrix(rnorm(n.genes*n.samples),
  nrow=n.genes)
am <- apply(data[, type=='A'], 1, mean)
bm <- apply(data[, type=='B'], 1, mean)
```

Some Simulations

```
av <- apply(data[, type=='A'], 1, var)
bv <- apply(data[, type=='B'], 1, var)
cv <- apply(data[, type=='C'], 1, var)

sp2.ab  <- ((an-1)*av + (bn-1)*bv)/
                (an+bn-2)
sp2.abc <- ((an-1)*av + (bn-1)*bv +
                 (cn-1)*cv)/(an+bn+cn-3)
```

Some Simulations

```
t.stat.ab  <- (bm - am)/
  (sqrt(sp2.ab)/sqrt(1/an+1/bn))
t.stat.abc <- (bm - am)/
  (sqrt(sp2.abc)/sqrt(1/an+1/bn))

p.val.ab <- sapply(t.stat, function(
  tv, df) {
   2*(1-pt(abs(tv), df))
  }, an + bn - 2)
p.val.abc <- sapply(t.stat,...
   , an + bn +cn - 3)
```

What Differences Are There?

None.

Added variability makes it harder to see stuff that is there, but not easier to see stuff that isn't there.

The benefits associated with more precision are linked to increased sensitivity.

Introduce some Differences

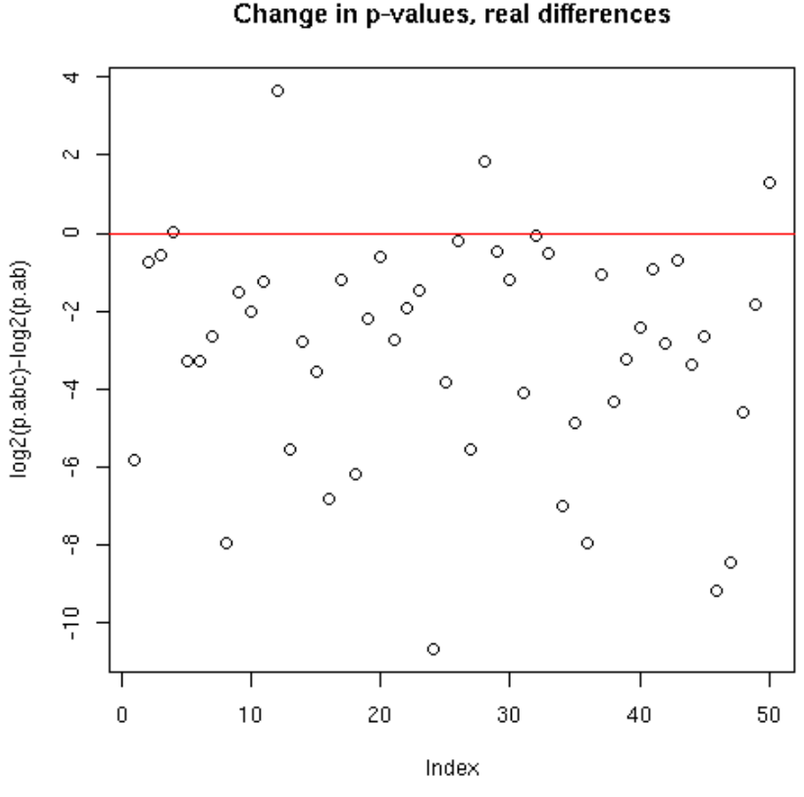
```
data[1:50,type=="A"] <-
    data[1:50,type=="A"] + 3;
```

recompute t-values and p-values

```
sum(p.val.ab  < 0.01); # gives 19
sum(p.val.abc < 0.01); # gives 45
sum(p.val.ab[1:50]  < 0.01); # gives  2
sum(p.val.abc[1:50] < 0.01); # gives 21
```

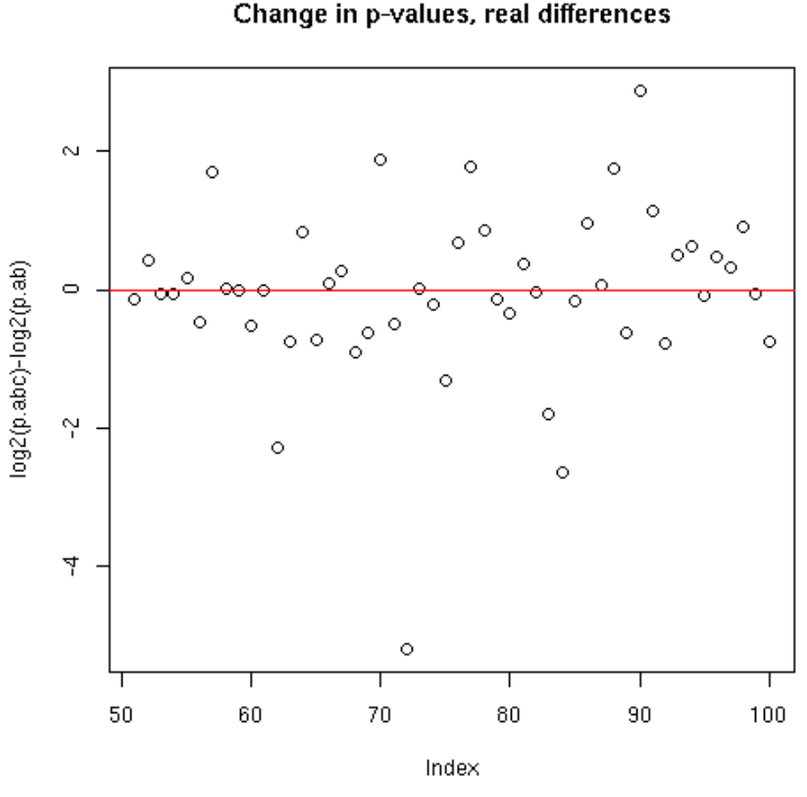
Plot P-Value Differences

```
plot(-log2(p.val.ab[1:50]) +
       log2(p.val.abc[1:50]), ... );
```



Plot P-Value Differences

```
plot(-log2(p.val.ab[51:100]) +
       log2(p.val.abc[51:100]), ... );
```



What Assumptions are We Making?

The variance structures do not change between the three groups (the means can be different).

We are already making this assumption implicitly with the two-sample t-test.

This assumption means that I would restrict the other groups used to those run about the same time, with the same chip lot, etc.

That the data looks approximately normal (work on the log scale).

Corrections

Rank tests also work.

Bonferroni still works just fine.

BUM still works just fine.

Empirical Bayes still works just fine.

Permutations?

Permute residuals from the null model

Another Extension: Chip Lot?

Say we have data from arrays from two different lots, 1 and 2, and that we have samples from groups A and B run on arrays from both lots. How should we look at this?

Well, we can still use a two-sample t-test (assuming run order was randomized), but this might break if there are big differences between lots.

(I'll assume for now that the number of samples in each group/lot combination is the same).

Some more Simulations

```
n.genes <- 2000
a1n <- 2;  b1n <- 2
a2n <- 2;  b2n <- 2
an <- a1n + a2n;  bn <- b1n + b2n;
n.samples <- an + bn;
type <- factor(rep(c('A', 'B'),
     times=c(a1n + a2n, b1n + b2n)))
group <- factor(c(rep(c('G1', 'G2'),
                      times=c(a1n,a2n)),
                      rep(c('G1', 'G2'),
                      times=c(b1n,b2n))));
```

Add Some Big Differences

```
data <- matrix(rnorm(n.genes*n.samples)
  nrow=n.genes)

data[,group=="G2"] <-
    data[,group=="G2"] + 8;

data[1:50,type=="A"] <-
    data[1:50,type=="A"] + 4;
```

Is this realistic? Can groups overshadow types?

How do we fit both type and group?

Start with an overall mean

measure deviations associated with type

measure deviations associated with group

```
mu <- apply(data,1,mean);
delta.type  <- apply(
    data[,type=="A"]-mu,1,mean);
delta.group <- apply(
    data[,group=="G1"]-mu,1,mean);
```

How do we fit both type and group?

fit the data, and sum the squared residuals

```
our.fit <- data;
our.fit[,type=="A" & group=="G1"] <-
  mu + delta.type + delta.group;
our.fit[,type=="A" & group=="G2"] <-
  mu + delta.type - delta.group;
our.fit[,type=="B" & group=="G1"] <-
  mu - delta.type + delta.group;
our.fit[,type=="B" & group=="G2"] <-
  mu - delta.type - delta.group;
```

Some numbers

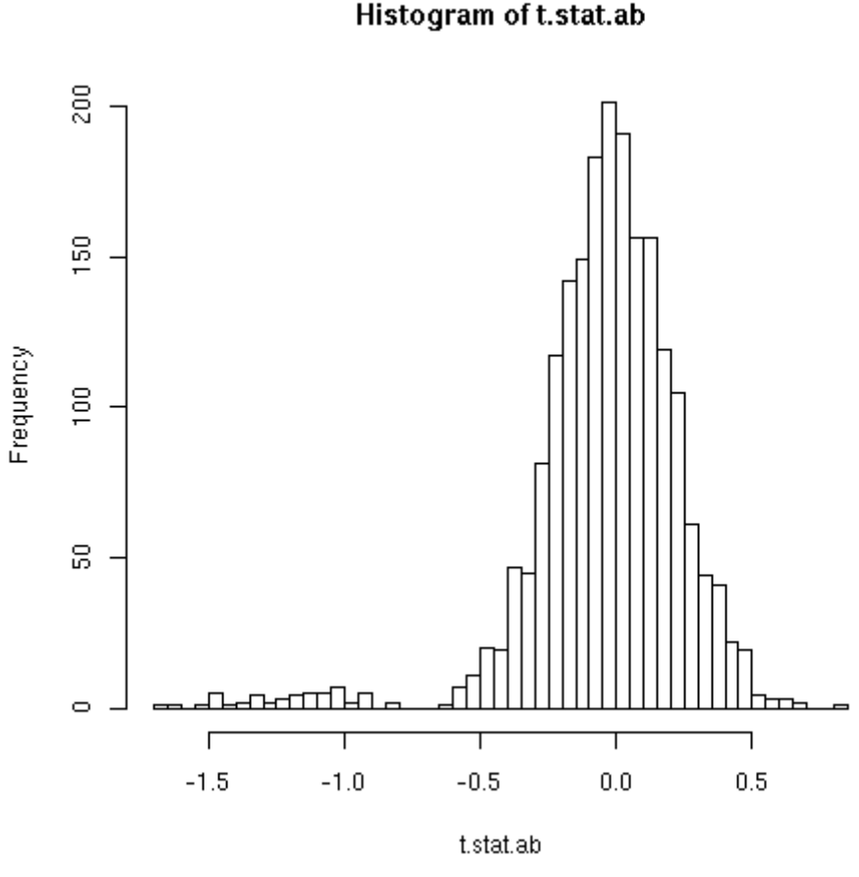
```
> our.resid <- data - our.fit;
> our.se <- sqrt(apply(our.residˆ2,
                       1, sum)/5);
> data[1,]
[1]  3.81  3.59 11.11 12.54
[5] -0.67 -0.17  7.05  8.95
> mu[1]
[1] 5.78
> delta.type[1]
[1] 1.99
> delta.group[1]
[1] -4.14
```

Some numbers

```
> our.fit[1,]
[1]  3.63  3.63 11.90 11.90
[5] -0.35 -0.35  7.93  7.93
> our.resid[1,]
[1]  0.18 -0.04 -0.79 0.64
[5] -0.32  0.17 -0.88 1.03
> our.se[1]
[1] 0.79
our.t.type <- delta.type[1]/
              (our.se[1]/sqrt(8));
```

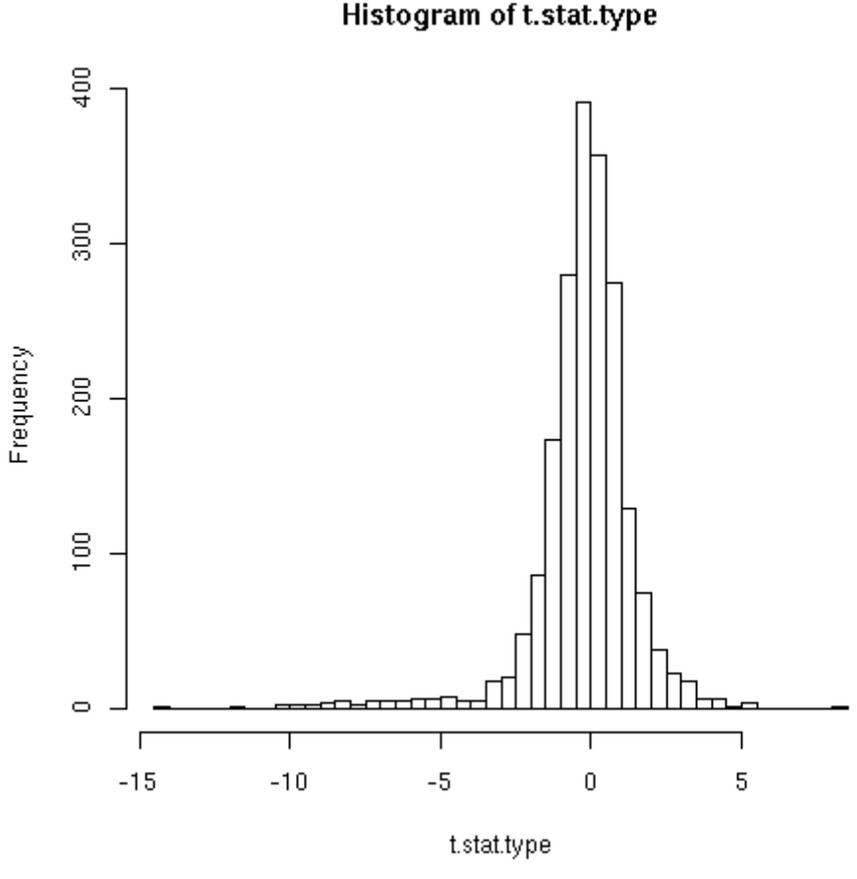
What do the t-stats look like?

```
hist(t.stat.ab,breaks=50);
```



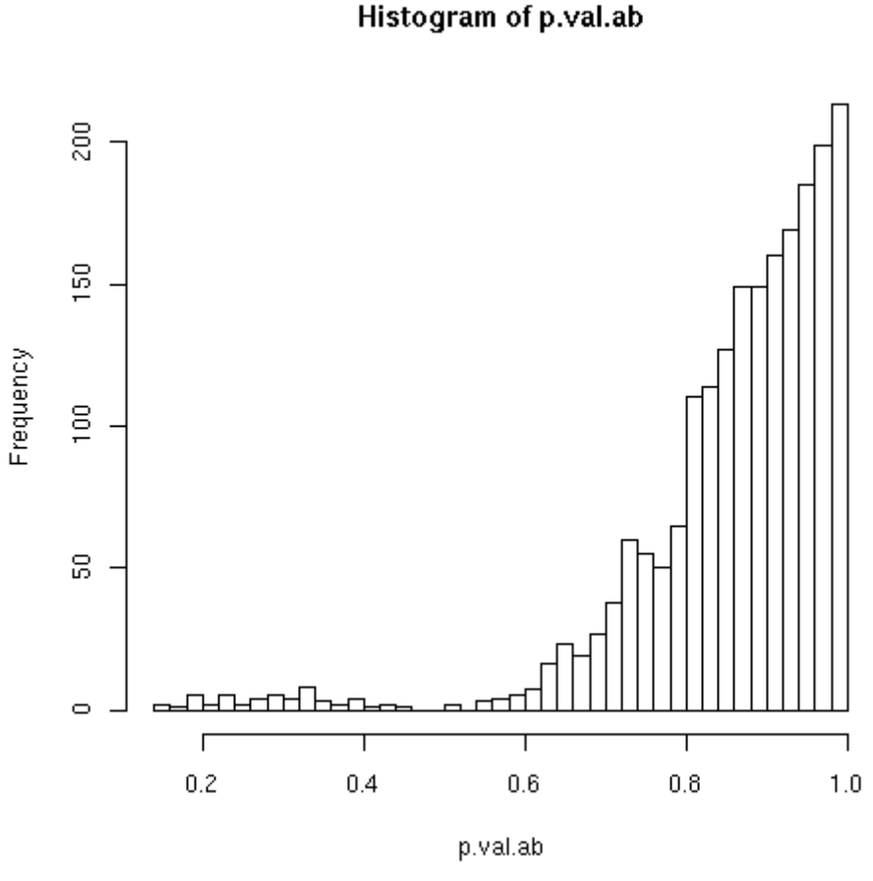
What do the t-stats look like?

```
hist(t.stat.type,breaks=50);
```



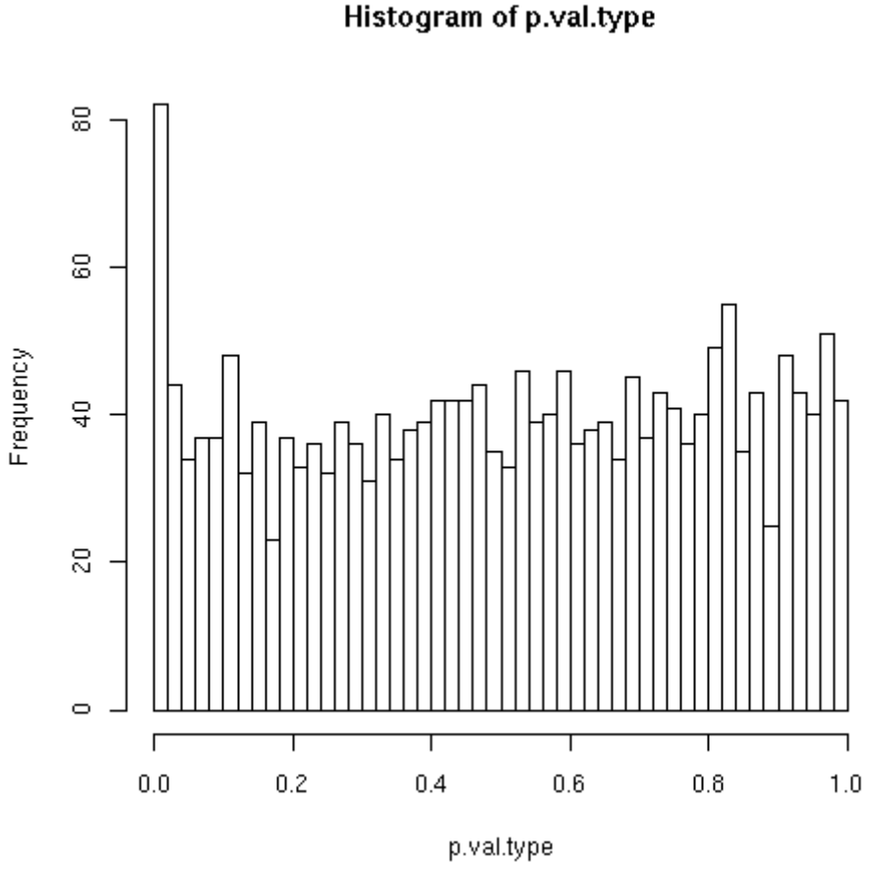
What do the p-values look like?

```
hist(p.val.ab,breaks=50);
```



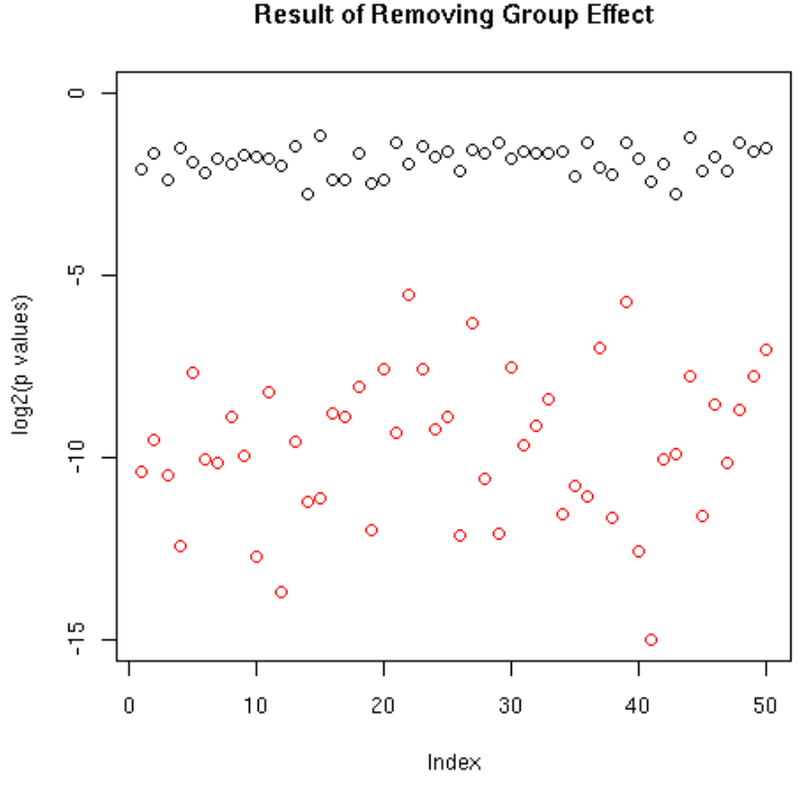
What do the p-values look like?

```
hist(p.val.type,breaks=50);
```



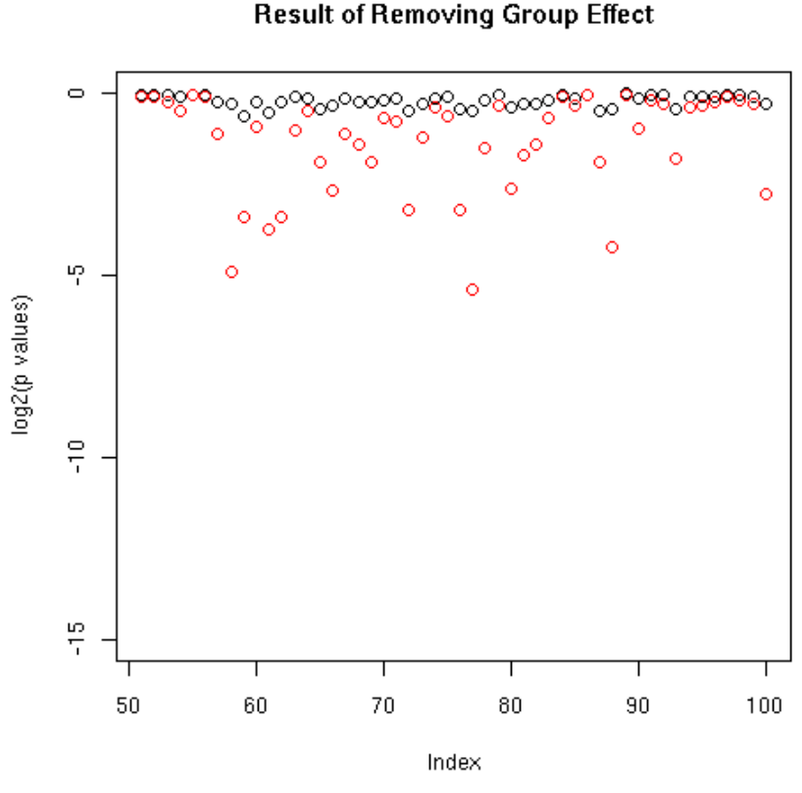
Changes When Different

```
plot(log2(p.val.ab[1:50]),ylim=c(-15,0),...)
points(log2(p.val.type[1:50]),col='red');
```



Changes When Different

```
plot(c(51:100),log2(p.val.ab[1:50]),...);
points(c(51:100),log2(p.val.type[1:50]),...)
```



Partitioning Variance: ANOVA

This general procedure of apportioning the observed variation to the effects that gave rise to it is known as the Analysis of Variance (ANOVA). It was introduced by R.A. Fisher in the 1920s.

Using other groups to stabilize the variance may not be that big a deal. Splitting off variation due to external causes before assessing our effect of interest can be vital.

ANOVA in R

```
our.lm.1 <- lm(data[1,] ~ type + group);
our.anova.1 <- anova(our.lm.1);
our.anova.1
Analysis of Variance Table

Response: data[1, ]
          Df  Sum Sq Mean Sq F value
type       1  31.557  31.557  52.008
                Pr(>F) 0.000799 ***
group      1 136.818 136.818 225.487
                Pr(>F) 2.372e-05 ***
Residuals  5   3.034   0.607
```

Replacing Data with Ranks

```
our.lm.1 <- lm(rank(data[1,]) ~ ...);
our.anova.1 <- anova(our.lm.1);
our.anova.1
Analysis of Variance Table

Response: rank(data[1, ])
          Df Sum Sq Mean Sq F value
type       1    8.0     8.0      20
                Pr(>F) 0.0065663 **
group      1   32.0    32.0      80
                Pr(>F) 0.0002911 ***
Residuals  5    2.0     0.4
```

Is this Legit?

Not really. The rank theory is based on shuffling just those values around, and requires somewhat larger sample sizes to get close to “normal”.

The more standard rank test for any difference at all here is the Kruskal-Wallis test. The p-values for this test are computed by permuting ranks and counting the possible sums.

Kruskal-Wallis Results

```
our.kw <- kruskal.test(rank(data[1,]) ~
                         type + group)
Kruskal-Wallis rank sum test

data:  rank(data[1, ]) by type by group
Kruskal-Wallis
    chi-squared = 1.3333, df = 1,
    p-value = 0.2482
```

Not enough samples here, so we need some assumptions.

An Extreme Case: Pairing

In many cases, we have data that are paired: treated/untreated, before/after, primary/metastasis (same patient), or case/control studies matched on a variety of factors.

In this case the math simplifies rather considerably, and we can use a simple one-sample t-test applied to the paired differences:

$$\frac{\bar{x}_A - \bar{x}_B}{sqrt(var(data[A] - data[B])/(n_A - 1))}$$

The Rank Equivalent: Signed Rank Tests

As with the ANOVA table discussed above, the paired t-test also has a rank analog, arrived at by ranking the differeces and applying a sign as A is greater than B or vice-versa. The sum of the positive ranks gives the test statistic.

```
wilcox.test(data[A],data[B],paired=TRUE);
```

Three Groups, Two Lots?

What if we have both scenarios at once?

Mutliple groups, and known external factors?

What is the general rule?

Including Covariates

The general extension of ANOVA is supplied by the linear model and regression. This was actually used above:

```
our.lm.1 <- lm(data[1,] ~ type + group);
```

where we are fitting the response (data[1,]) as a function of the covariates at hand (type and group). The final significance value is that associated with the effect of interest in the full model.

Some Standard Factors

What things might we include as explanatory covariates?

- chip lot
- chip
- dye
- run date/order

The Broader Theme: Modelling

If we know that effects other than the ones we're interested in are likely to be present, it is generally worthwhile to recast our test to explicitly incorporate (and hopefully factor out) these other effects.

This is the idea of modelling the data.

Of course, we can't model everything. When we can't model it, randomize to balance it!