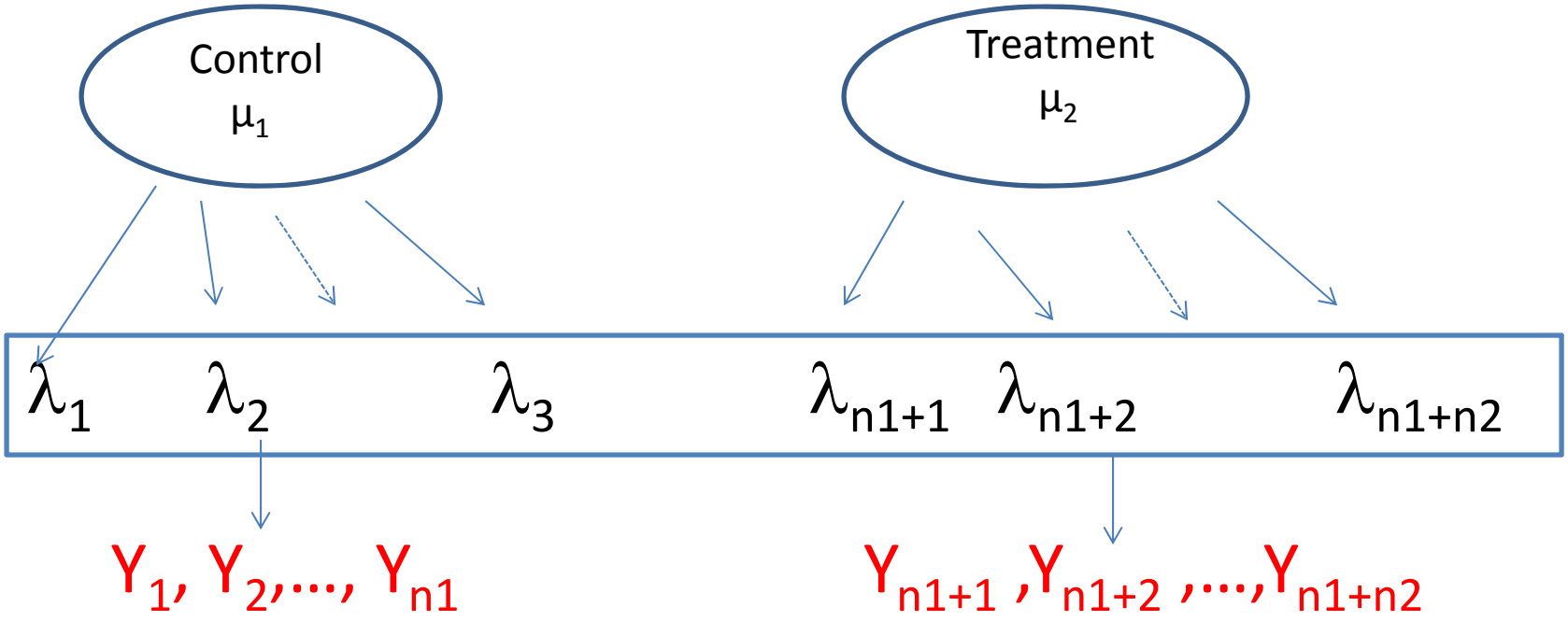
Differential Expression in RNA sequencing

Differential Expression

- Question of interest: $\mu_1 = \mu_2$???



Modeling the counts Y

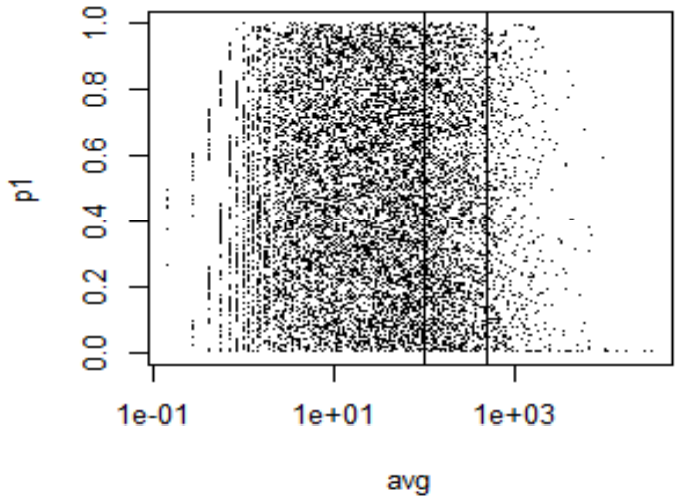
- Sequencing alone is close to Poisson counting

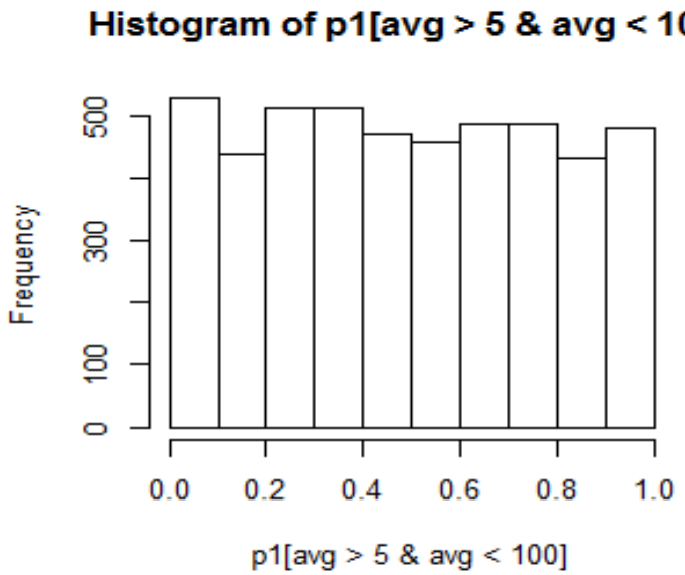
$$Y_{gi} \mid \lambda_{gi} \sim \text{Poisson}(\lambda_{gi})$$

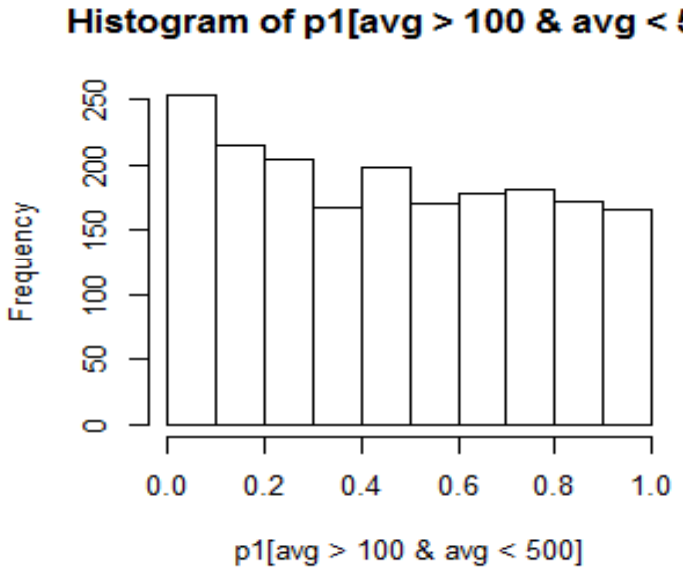
- Question : Can you check this?
  - Chi square test

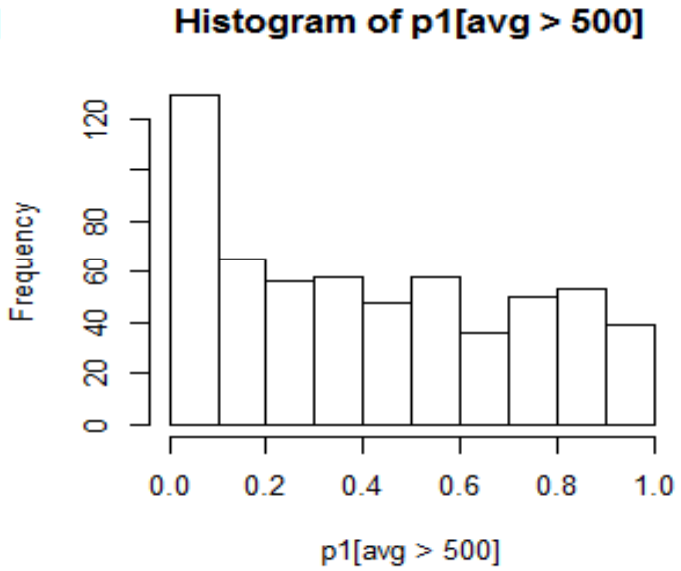
A quick check on maqc data

```
load("maqc_eset.RData")
library(Biobase)
x=exprs(maqc.eset)
x1=x[,1:7]##
T1=colSums(x1)
p1=apply(x1,1, function(xxx)
chisq.test(rbind(xxx,T1-xxx))$p.value)
avg=rowSums(x1)/sum(T1)*median(T1)
par(mfrow=c(2,2))
plot(cbind(avg,p1)[avg>0,],pch=".",log="x")
abline(v=c(100,500))
hist(p1[avg>5&avg<100])
hist(p1[avg>100&avg<500])
```


p1
0.0
0.2
0.4
0.6
0.8
1.0
1e-01
1e+01
1e+03
avg


Histogram of p1[avg > 5 & avg < 100]
Frequency
0
100
300
500
0.0
0.2
0.4
0.6
0.8
1.0
p1[avg > 5 & avg < 100]


Histogram of p1[avg > 100 & avg < 500]
Frequency
0
50
100
150
200
250
0.0
0.2
0.4
0.6
0.8
1.0
p1[avg > 100 & avg < 500]


Histogram of p1[avg > 500]
Frequency
0
20
40
60
80
120
0.0
0.2
0.4
0.6
0.8
1.0
p1[avg > 500]

Modeling the counts Y

- $Y_{gi} | \lambda_{gi} \sim \text{Poisson}(\lambda_{gi})$
- $\lambda_{gi} \sim \text{Gamma}(S_{gi}\, \mu_{g,k(i)}, \Phi_g)$
  - $\mu_{g,k(i)}$ represents the mean expression level for gene g under condition k(i)
  - S_{gi} is a normalization offset (constant) representing sequencing bias between genes and samples
  - Φ_g represents biological variation between replicates under the same condition
  - Φ_g is closely related to “gene-specific variance” in microarray data in log scale

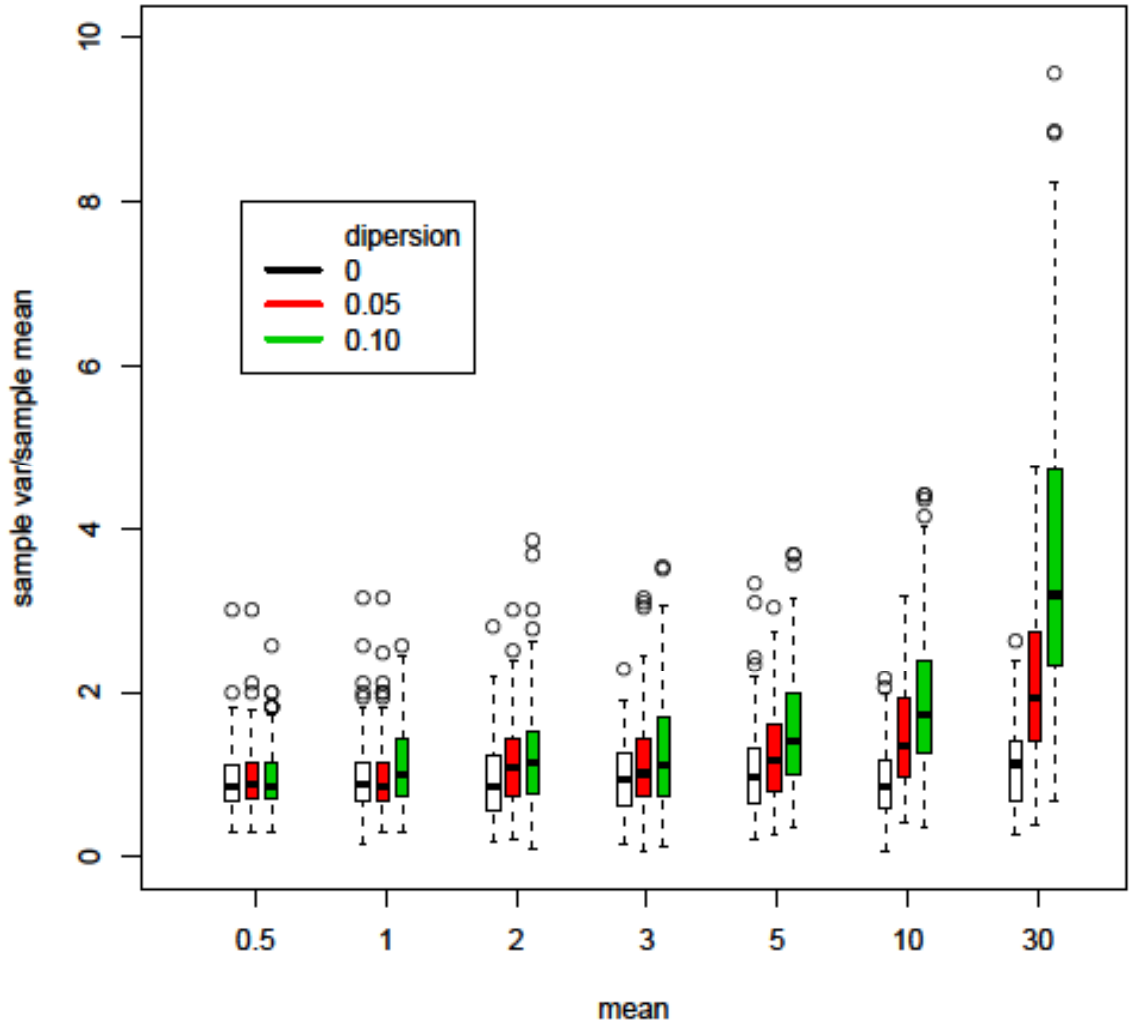
Coefficient of variation

- CV represents variation relative to mean

  CV= SD/mean

- $CV^2(y_{gi}) = 1/\mu_{g,k(i)} + \Phi_g$
  - For highly expressed genes, $1/\mu_{g,k(i)}$ is negligible compared to Φ_g, so variance mostly determined by biological variation
  - For modestly expressed genes, the variance due to Poisson counting can not be ignored.

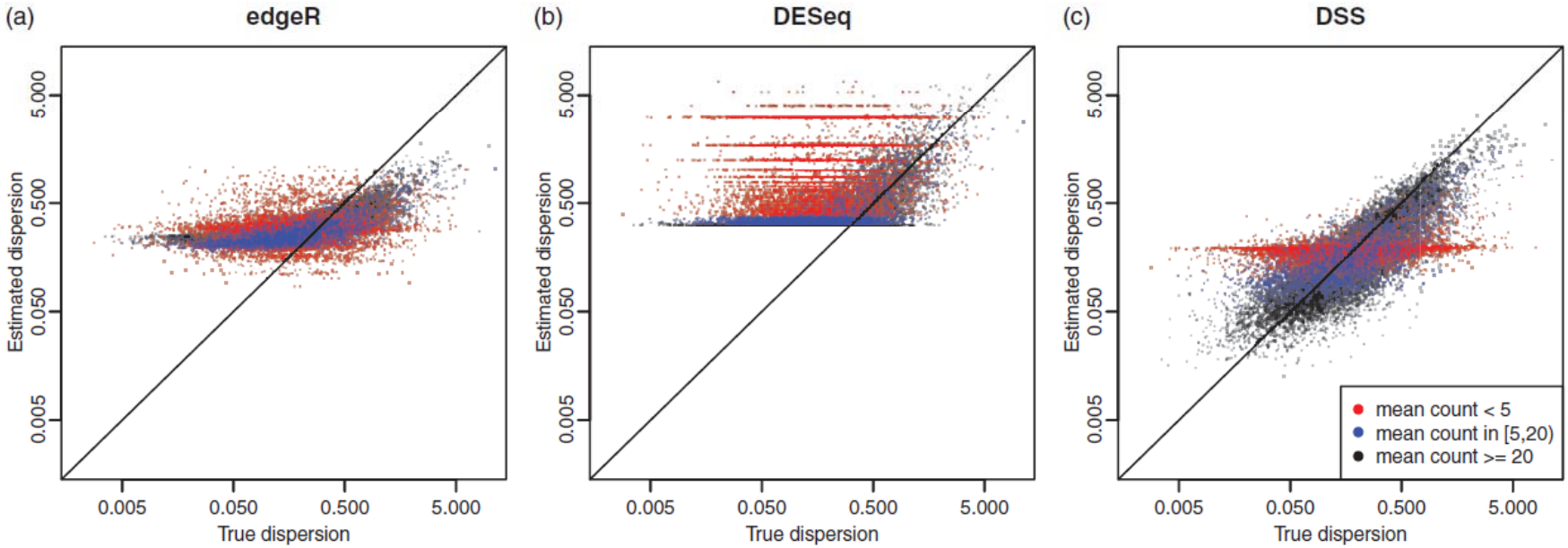
Is over-dispersion observable?



- Φ_g can be estimated from data when you have many replicates
- The estimate is not very good when you have few replicates, or when the gene is not expressed high
- Several methods of shrinkage tries to estimate Φ_g by borrowing information across genes

Some examples of dispersion shrinkage

- Common dispersion, complete shrinkage global ϕ
  - Robinson and Smyth (2008)
- Dispersion depends on mean expression
  - Anders and Huber (2010)
- Partial shrinkage towards a common prior (conditioning on mean)
  - Robinson and Smyth (2007)
- Shrinkage based on a lognormal prior
  - Wu et al (2012)


(a)
edgeR
(b)
DESeq
(c)
DSS
Estimated dispersion
True dispersion
0.005
0.050
0.500
5.000
mean count < 5
mean count in [5,20)
mean count >= 20

Testing and inference in two-group comparison

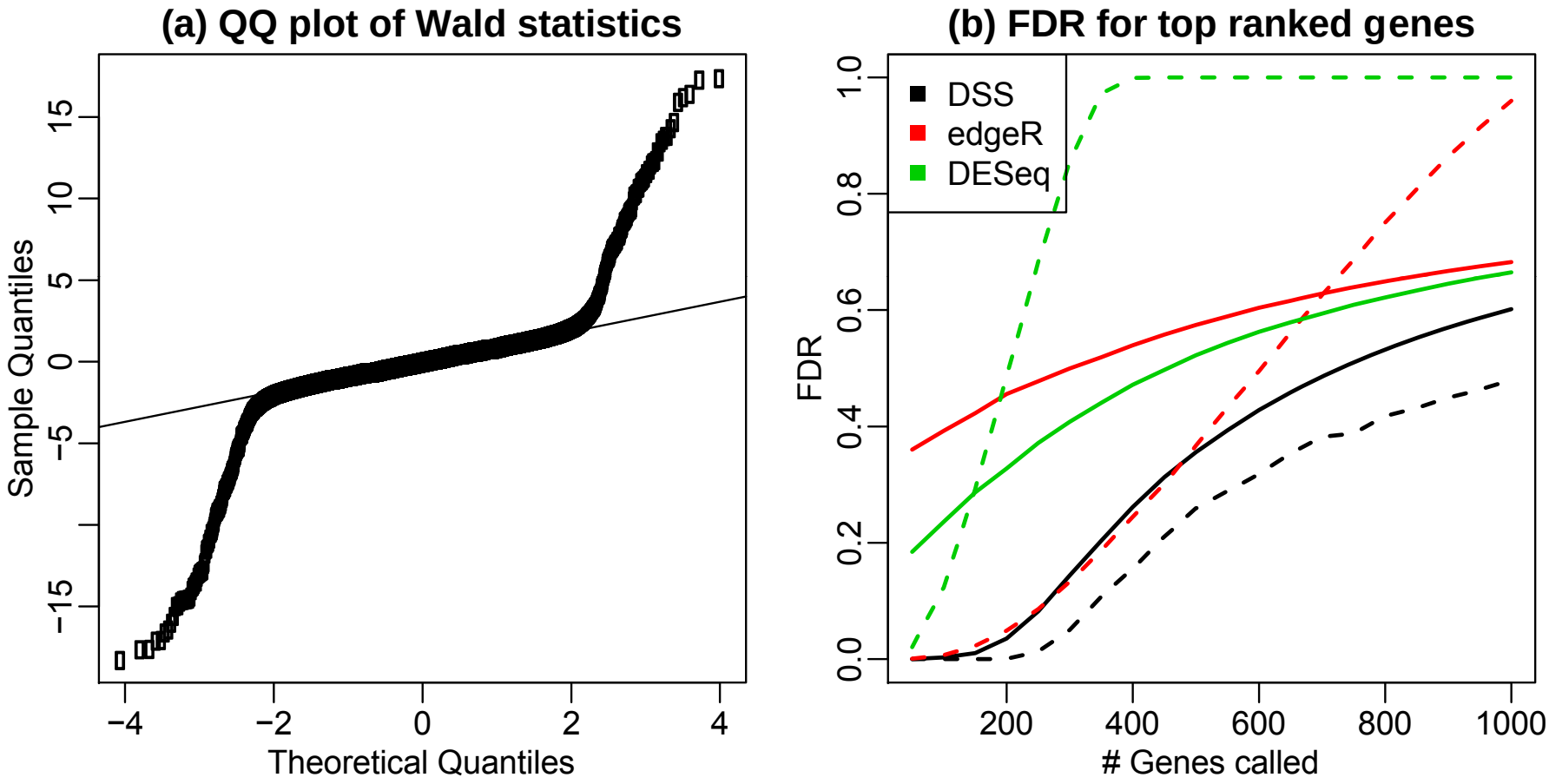
- Exact test based on NB distribution

- Wald test: $t_g = \frac{\hat{\mu}_{g,1} - \hat{\mu}_{g,2}}{\sqrt{\hat{\sigma}^2_{g,1} + \hat{\sigma}^2_{g,2}}}$

  – With dispersions, variances can be computed according to NB distribution: $var = \mu + \mu^2\phi$

  – The variance for $\hat{\mu}_{g,1}$ is: $\hat{\sigma}^2_{g,1} \equiv \frac{1}{n_1^2}\left[\hat{\mu}_{g,1}\left(\sum_{j:k(j)=1}\frac{1}{s_j}\right) + n_1\hat{\mu}^2_{g,1}\tilde{\phi}_g\right]$

Statistical inference



Case studies

- edgeR as example
  - Not necessarily the best but very well documented user guide for beginners
  - In some simulations found to be too conservative in FDR estimation
  - Do you own simulation for your situation

One factor, two or more groups

```
> x <- read.delim("fileofcounts.txt",row.names="Symbol")
#### depends on how your data file is formatted, this may be
   slightly different
> group <- factor(c(1,1,2,2))
## two groups: first two sample in one group
> y <- DGEList(counts=x,group=group)
> y <- calcNormFactors(y)
## normalization factor estimated and stored in “y”
> y <- estimateCommonDisp(y)
## overall scale of dispersion
> y <- estimateTagwiseDisp(y)
## how each gene (tag) varies
> et <- exactTest(y)
## statistical test
> topTags(et)
## find a top list. Similar to topTable
```

Matched pairs

```
Pair <- factor(c("A", "A", "B", "B", "C", "C"))
Tissue <- factor(c("N","T","N","T","N","T"))
design <- model.matrix(~Pair+Tissue)
design
  (Intercept) PairB PairC TissueT
1           1     0     0       0
2           1     0     0       1
3           1     1     0       0
4           1     1     0       1
5           1     0     1       0
6           1     0     1       1
```

```
y <- estimateGLMCommonDisp(y, design, verbose=TRUE)
y <- estimateGLMTrendedDisp(y, design)
y <- estimateGLMTagwiseDisp(y, design)
#### unlike linear models that estimate the residual
  variance in the regression, dispersion is
  separated estimated here
fit <- glmFit(y, design)
### fit generalized linear models instead of LM
lrt <- glmLRT(fit)
### use likelihood ratio test instead of (moderated
  F-test)
topTags(lrt)
### similar to top tables.
```