Statistical hypothesis testing

Testing for differential gene expression

- In simplest case, performs the equivalent of a t-test between two groups (e.g. disease vs control)
- We can simulate some gene expression data with an R function to understand the idea of differential expression and significance

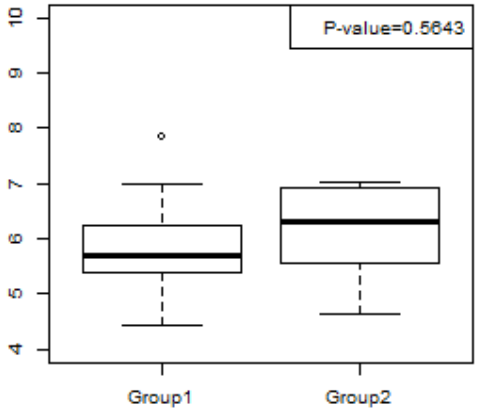
```
sim.data <- function(mean1, sd1, mean2, sd2)
{
      x <- round(rnorm(10, mean1, sd1),4)
      y <- round(rnorm(10, mean2, sd2),4)
      t.out <- t.test(x, y)
      print(t.out)
      p <- round(t.out$p.value, 4)
      boxplot(x, y, names=c("Group1", "Group2"), ylim=c(0,12))
      legend("topright", lty=0, legend=paste0("P-value=", p), cex-0.8)
      print(paste0("P-value of t-test: ", p))
  }
```

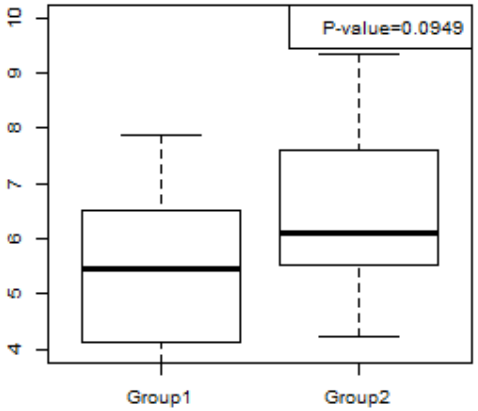
Simulating differential expression

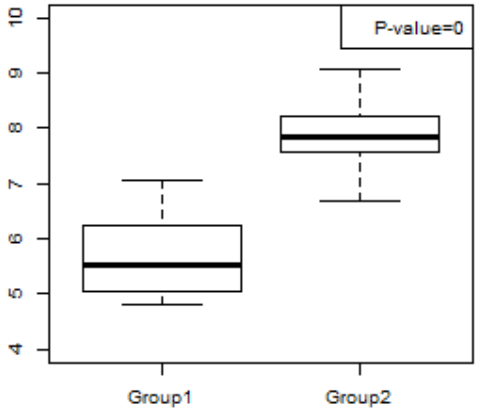
- Run this function for a few scenarios to get a feel for what affects the p-value (remember that $p<0.05$ indicates a statistically significant difference in means between 2 groups)

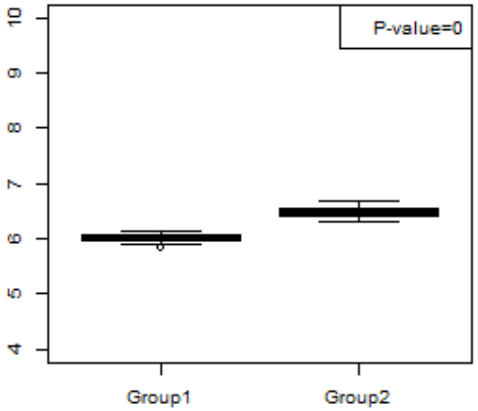
```
par(mfrow=c(2,2)) # to display 4 plots
sim.data(6,1,6,1) # group means equal
sim.data(6,2,6.5,2) # small difference in group means, relatively high variability
sim.data(6,1,8,1) # large difference in group means
sim.data(6,0.4, 6.5,0.4) # small difference in group means, but also very low variability within groups
```

Examples of simulated data









Fold changes are only half the story

- Simulations 2 and 4 have the same difference in means, but high and low variance within groups respectively
- The effect is clear from the boxplots: simulation 4 produces data tightly clustered around the mean while simulation 2 has much larger spread and considerable overlap between the groups
- Though the estimated fold changes will be similar, the p-values should be very different (only likely to be significant in the latter)
- Ranking on fold change can therefore be misleading
- The fold change does not tell you anything about the variability of the gene, which can dramatically affect whether such a change might occur by chance or not – for this, p-values are needed

T-test and statistical significance

- Standard (2-sample) t-test
- $t = \frac{\mu_1 - \mu_2}{SE(\mu_1 - \mu_2)}$
- Standard error is larger with smaller sample sizes and/or larger variance (increased unreliability in the estimate of the difference of means)
- Thus, a large difference between the means (fold change) and a small associated SE will give large values for the t-statistic, and corresponding small p-values
- Rule of thumb: $|t| > 2$ likely to be significant
- Note that the same or similar fold change can have quite different associated p-values depending on the variability of the gene
- The sample variance is used in the calculation of t-statistic, and this can be a problem when sample sizes are small – often the case for gene expression studies

Variance and Sample Size

- Simulation exercise to understand the relationship between sample size and estimated value of variance (see p42-43 of Crawley's textbook)
- Sample from a normal distribution (mean=10, sd=2) for sample size between n=3 and n=31, each done 30 times and plot the resulting variance estimates (var=sd^2 = 4)
- How might we go about writing some R code to do this simulation?
- Let's think about each step in turn…

Michael J Crawley: Statistics, An Introduction using R

Variance and Sample Size

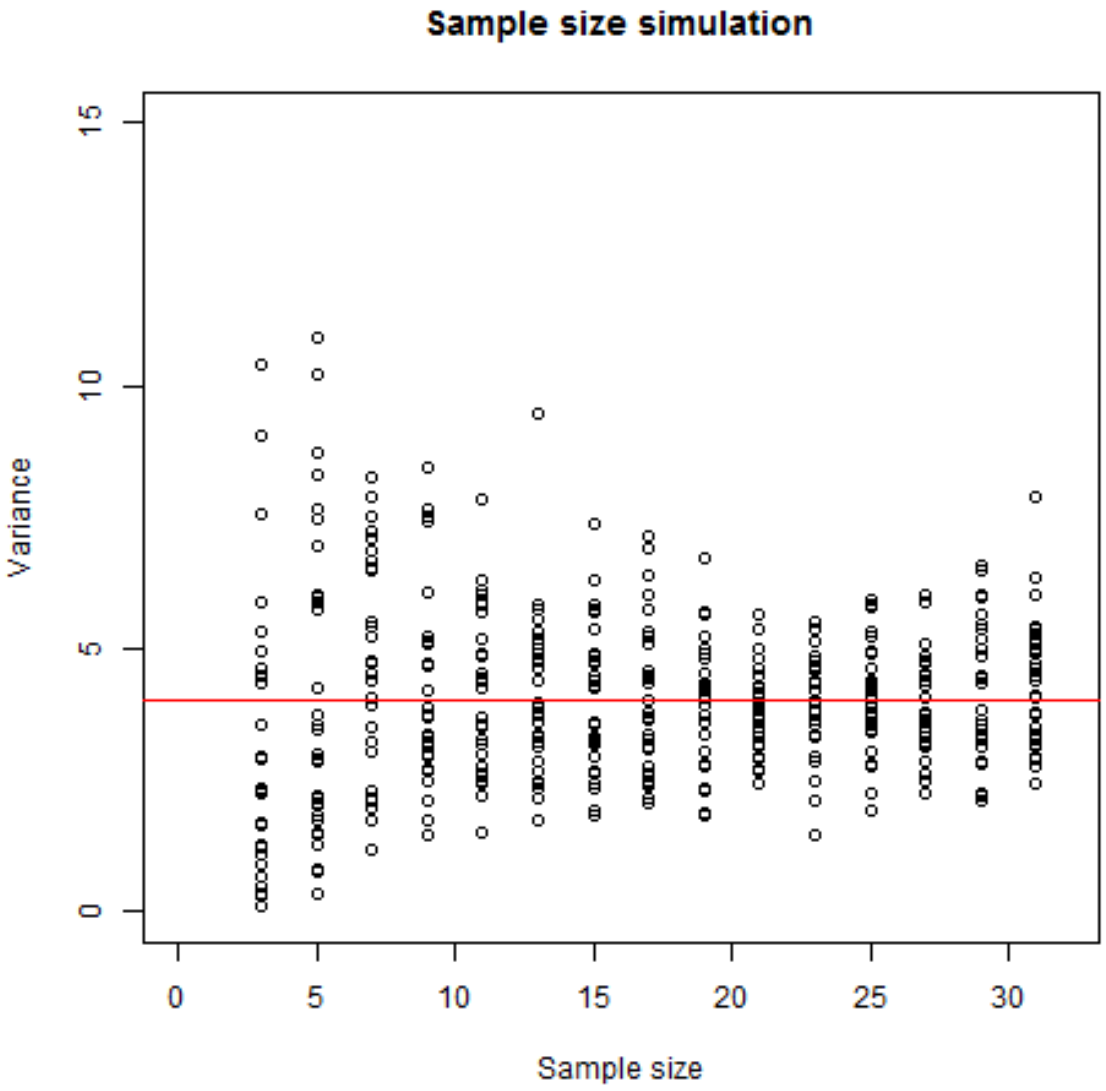
- Simulataion exercise to understand the relationship between sample size and estimated value of variance (see p42-43 of Crawley's textbook)
- Sample from a normal distribution (mean=10, sd=2) for sample size between n=3 and n=31, each done 30 times and plot the resulting variance estimates (var=sd^2 = 4)
- How might we go about writing some R code to do this simulation?
- Let's think about each step in turn…
  - Draw data from specified distribution for each sample size (rnorm)
  - Need to do this 30 times per sample size – suggests loop
  - Need to compute the variance of each sample drawn
  - Plot the results

Michael J Crawley: Statistics, An Introduction using R

Variance and Sample Size

```
png("Sample_size_simulation_plot.png")

plot(c(0,32), c(0,15), type="n", xlab="Sample size",
ylab="Variance", main="Sample size simulation")

for(df in seq(3,31,2)){
for(i in 1:30){
x <- rnorm(df, mean=10, sd=2)
points(df, var(x)) }}

abline(4,0, col="red")

dev.off()
```

Results of simulation



Gene expression analysis using R/BioConductor

Limma package (from Bioconductor)

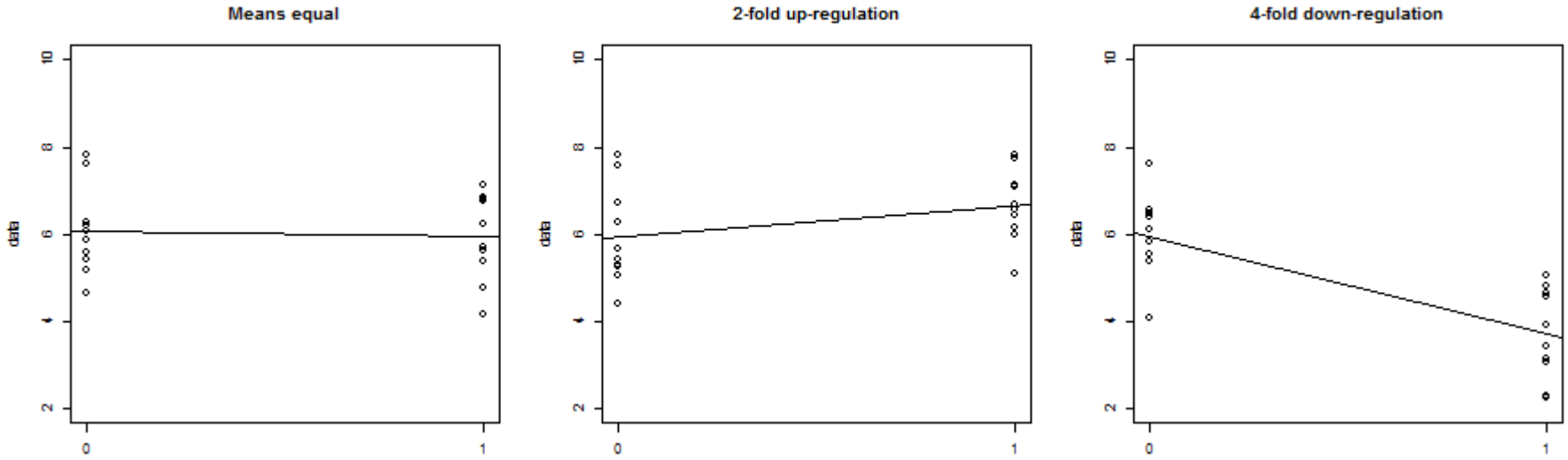
- limma: Linear models for microarray data
- http://bioconductor.org/packages/release/bioc/html/limma.html
- Originally developed over 15 years ago to handle microarray data (including preprocessing and data analysis) and provides a comprehensive framework for gene expression data analysis
- Widely used and gold standard in the field, developed by Gordon Smyth and colleagues at Walter and Eliza Hall Institute (WEHI), Melbourne, Australia
- Some aspects now obsolete e.g. methods for 2-colour microarrays but has evolved with the field of transcriptomics and now able to input RNA-Seq data to limma (more on this later) if suitably transformed beforehand
- Introduces some fundamental concepts for analysing gene expression data in R
- Implements standard statistical methods (linear regression) but with additional features tailored to gene expression experiments

Using limma

- Despite extensive and regularly updated documentation (current userguide is 145 pages!), limma may be difficult to use correctly for those new to statistical data analysis on large-scale data
- Today, we will look at how to set up statistical analyses in R using limma and understand the details and pitfalls to run reliable gene expression analysis
- Although the approach and some terminology may be new, the basic idea is testing for differences in gene expression between groups, with associated statistical evidence (p-values) for any changes
- Limma can handle arbitrarily complex experimental designs but we will focus mostly on a simple comparison of two groups

Linear model approach

- Tests whether the slope of the regression line is significantly different from zero (equivalent to a t-test in fact)
- Explanatory variable often categorical (disease vs control, treatment vs control, mutant vs wildtype etc)



- Slope of the regression line is equal to the difference in means of the two groups (if model has intercept term)
- The number of samples and variability in the data determines the confidence in the coefficient estimate (standard error)

Testing for differential expression

- The difference between 2 groups can be estimated directly if an intercept term is included in the model (estimates mean expression in control or reference group). The slope of the regression line indicates the difference between this group and a second group:
  - if they are similarly expressed, the line will be almost flat and the regression coefficient close to zero
  - if a gene is up-regulated in the second group, the coefficient will be positive (and quite large) – adding this amount to mean baseline expression gives the mean expression in the second group
  - Similarly, if a gene is down-regulated in the second group, the coefficient will be large and negative (essentially need to subtract from the baseline level to get the mean expression of the second group)
- Again, the number of samples in each group and the variability determine the significance of the difference
- Ranking on p-value, rather than logFC, identifies the most reliable results

Using limma effectively

- Functions in limma perform all the statistical calculations behind the scenes
- A linear model is fitted to each gene separately, but even analysing >20,000 genes is done very quickly – highlights the efficiency of R for these kind of applications
- Rather than the underlying statistical analysis, the user needs to focus on making sure the input to limma functions is correct, and especially that the experimental design is correctly defined
- Including relevant explanatory variables enables limma to detect differentially expressed genes with greater power
  - e.g. batch, litter or other similar effects may explain some variability in the data, which limma can account for
  - Age, gender and other demographic or clinical phenotype data can also be included in the model (so long as not confounded with experimental groups)

eBayes to improve variance estimation

- Empirical method to make more robust inferences of variance for individual genes by 'borrowing' information across genes
- Shrinks the variance estimates towards a common value
  - High variance genes get their estimates squeezed down a bit
  - Low variance genes get their estimates squeezed up towards the typical variance seen for genes of similar mean expression
- Reduces the probability of calling small fold changes significant if the sample variance grossly under-estimates the true variance (likely for some, maybe many, genes among 20,000 given the typically small sample sizes of transcriptomics studies)

Sample limma code for simple experiment

- Input – normalised expression data (log2 scale)

```
library(limma)
eset <- read.table("Normalised_expression_data.txt",
sep="\t", header=T, row.names=1)
Group <- factor(c("WT", "WT", "Mu", "Mu", "Mu"))
design <- model.matrix(~Group)
colnames(design) <- c("WT", "MUvsWT")
fit <- lmFit(eset, design)
fit <- eBayes(fit)
topTable(fit, coef="MUvsWT", adjust="BH")
```

- Output –list of genes ranked by evidence for differential expression

Limma – analysis made simple (if used correctly)

- This shows that running differential expression analysis for simple experiments is not too difficult and we will run some analysis later on a dataset with 3 different cancer types
- But is also deceptively simple and it's important to understand what each step is doing – there are many ways to get it wrong and only a couple of ways to get it right (we'll see alternative models that output the same results in a minute).
- Also important to know how to check that limma has done what you want it to do.

RNA-Seq analysis packages

- A variety of packages for processing and analysing RNA-Seq data were developed to handle count-based expression data and a myriad of sequencing-related effects and biases (longer genes generate higher counts, amplification biases related to sequence composition, library composition/complexity and so on).

- Tailored methods required for every step including alignment, summarising gene/transcript counts, normalisation, testing for differential expression and pathway/enrichment analysis

- For differential expression analysis, the dilemma was whether to develop novel methods and work directly with the counts, or transform the data in a way to meet the assumptions of the existing methods for microarray data

edgeR

- Analagous steps to limma but uses different statistical models for testing for differential expression
  - Computationally efficient and able to analyse complex experimental designs
  - Biological variability between samples increases the variance in the counts - negative binomial models fitted
  - Estimation of biological variability (dispersion) performed empirically
  - Like limma, borrows information across genes to improve estimates from small sample sizes

Limma-voom

- Alternative to count-based models is suitably transforming the counts and using existing standard statistical approaches
- Limma developers also did this so can use limma to analyse RNA-Seq data too!

Background to transcriptomics

- Microarray technology (1990s onwards): based on pre-designed short oligonucleotide sequences, or probes, hybridising to complementary target sequences (genes) – generate fluorescent intensity signal (continuous data that after preprocessing can be considered approximately normally distributed).

- RNA-Seq (~2008 onwards): next generation sequencing approach. Library of cDNA fragments prepared from RNA and sequenced. Generates count data (number of reads mapping to a given gene), requiring statistical models suitable for count data (e.g. negative binomial model as implemented in edgeR or DESeq)