Tutorial: Statistical Hypothesis Testing

The purpose of this tutorial is to introduce some relevant statistical concepts when testing for differential gene expression. We will use short exercises in R to illustrate how the magnitude of the fold change, variability and sample size all contribute to the overall significance of the result. In the simplest case, differential expression analysis performs the equivalent of a t-test between two groups (e.g. disease vs control) to compare the mean expression level in one condition to another.

Exercise 1 – Fold Changes and P-values

Open a new R session and copy/paste the command below (red text) to load some pre-written code ready for making some plots later. Note, after executing this command R will simply return to the prompt (indicated by the > symbol on a new line in the R console window). No output is expected until we enter some more commands (on next page).

```
source("http://www.well.ox.ac.uk/bioinformatics/training/RNASeq_Nov2018/Day2_271118/Files_and_scripts/Simulation_exercise1.R")
```

The following code shows the contents of the script that was loaded into R by the command we ran (this is for information only, you do not need to enter the lines in italics into R yourself).

```
sim.data <- function(mean.x, sd.x, mean.y, sd.y)
{
        x <- round(rnorm(100, mean.x, sd.x),4)
        y <- round(rnorm(100, mean.y, sd.y),4)
        t.out <- t.test(x, y)
        print(t.out)
        p <- round(t.out$p.value, 4)
        boxplot(x, y, names=c("GroupX", "GroupY"), ylim=c(0,12))
        legend("topright", lty=0, legend=paste0("P-value=", p), cex-0.8)
        print(paste0("P-value of t-test: ", p))
}
```

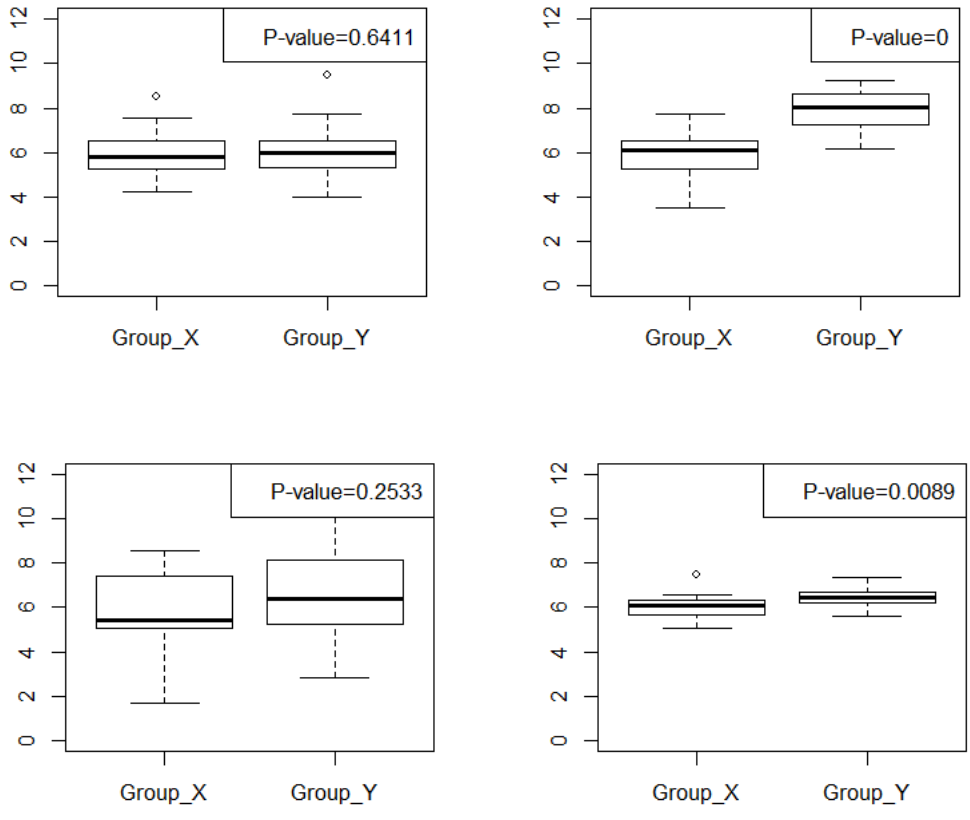
This is a short function named 'sim.data' that simulates some data from a normal distribution for two groups (which we have named x and y). We can modify the mean and standard deviation of the underlying distribution for each group separately using the four specified arguments mean.x, sd.x, mean.y and sd.y. It then uses R's in-built `t.test` function to test if there is a significant difference in the group means. This exercise simulates different scenarios to illustrate whether a significant p-value is likely to be obtained or not. You should get an idea of how both the magnitude of the difference between the group means and variability of the data influences the p-value.

Having defined our customised function, we can now use it in R just like any in-built function. To make it do something, we need to call the function, and provide the necessary arguments (we decide what values they take). We will run it 4 times with different conditions to get a feel for how the resulting p-value is affected (remember that p<0.05 indicates a statistically significant difference).

Enter the following commands one after the other (note everything after the # is ignored by R and is just a comment for us):

```
par(mfrow=c(2,2)) # to display 4 plots simultaneously, returns to prompt

sim.data(6,1,6,1) # both groups sampled from the same underlying
distribution, makes 1st plot

sim.data(6,1,8,1) # large difference in means, makes 2nd plot

sim.data(6,2,6.5,2) # small difference in means and high variability, makes
3rd plot

sim.data(6,0.4, 6.5,0.4) # small difference in means and low variability,
makes 4th plot
```

You should obtain plots similar to those below (note each person's simulation results will be slightly different because we are randomly sampling each time).



Take a few minutes to consider the resulting p-value in each case – which ones are significant? Discuss with a neighbour to make sure you understand the reason for each result.

T-tests and Statistical Significance

The first exercise illustrates that the fold change is only half the story - simulations 3 and 4 have the same difference in means, but high and low variability within groups respectively. The effect is clear from the boxplots: simulation 3 has much larger spread and considerable overlap between the groups while simulation 4 produces data tightly clustered around the mean. Though the estimated fold changes will be similar, the p-values should be very different (only likely to be significant in the latter). 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, and ranking genes on fold change alone can be misleading.

For a standard (2-sample) t-test, the t-statistic is defined as:

$$t = \frac{\mu_1 - \mu_2}{SE(\mu_1 - \mu_2)}$$

where the numerator is the difference in means, and the denominator is the standard error of the difference in the two means (more details below). Large values for the t-statistic (positive or negative, in the tails of the t-distribution) are evidence that the observed difference in means is unlikely to have occurred by chance, under the null hypothesis that the two group means are equal - rule of thumb is $|t| > 2$ likely to be significant.

The numerator is large when the difference between the means is large, and a relatively small value in the denominator will contribute further to a large value for the t-statistic overall, and correspondingly small (significant) p-values. The denominator is the **standard error of the difference between the group means** - this is a measure of the unreliability in the estimate of the difference of means and is larger when the sample size is small and/or there is higher variability in the data. It is the denominator term that explains why a similar fold change for two different genes can have quite different associated p-values; the calculation of the standard error includes the pooled sample variance estimate – when this is high, the SE increases, decreasing the t-statistic, and vice versa when the variance estimate is low.

Exercise 2 – Sample Size and Estimates of Variance

One of the problems of a typical gene expression experiment, whether using microarrays or RNA-Seq, is that the sample sizes are usually small (maybe 3-6 replicates per condition), and this makes the empirical estimate of the sample variance very unreliable. This next exercise illustrates the extent to which this could become a problem when analysing the data. It explores the relationship between sample size and variance estimates, and is taken from the textbook 'Statistics, An Introduction using R' by Michael J Crawley, p42-43.

We want to sample from a normal distribution (mean=10, sd=2) for sample sizes between n=3 and n=31, each done 30 times and plot the resulting variance estimates (var=sd^2; theoretical value 4)

How might we go about writing some R code to do this simulation?

Let's think about each step in turn…

- Need to draw data from specified distribution for each sample size (rnorm)
- Need to do this 30 times per sample size – suggests a loop would be useful
- Need to compute the variance of each sample drawn
- Finally, we need to plot the results

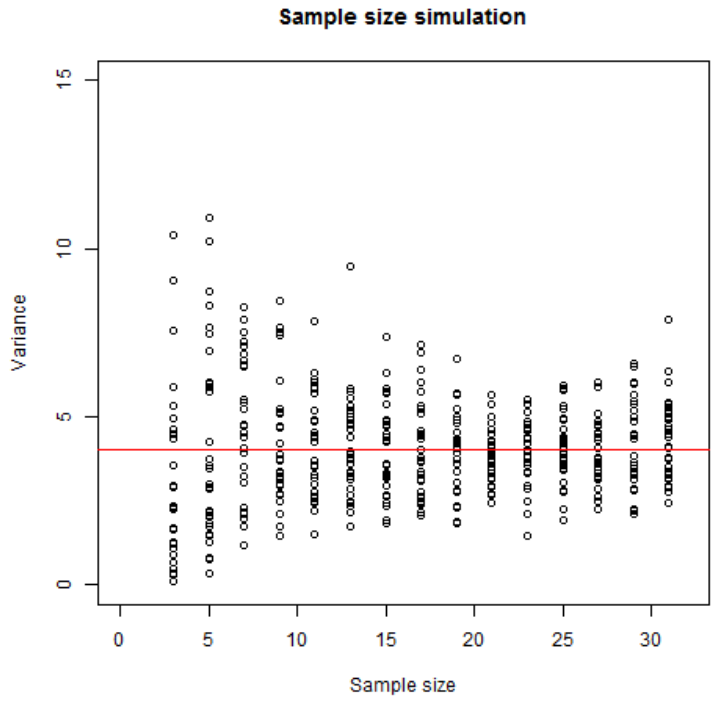
The code for a function to do this is in the file named Simulation_exercise2.R; you can load it directly into R with the following command:

```
source("http://www.well.ox.ac.uk/bioinformatics/training/RNASeq_Nov2018/Day2_271118/Files_and_scripts/Simulation_exercise2.R")
```

Again, the code is shown below for information only and you don't need to enter the following commands:

```
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")
```

When we run the code, it should produce a plot similar to this one:



If the true variance (which we know to be 4) is being well-estimated by the sample, the circles will fall on or very close to the red line. But a small sample is less likely to be representative of the population, and can produce very poor estimates. Note how the sample estimates deviate most from the red line (the known variance of our simulated distribution) when the sample size is less than 10, and tend to cluster more closely around it with the larger sample sizes.

Recall that the pooled sample variance estimate is used in the calculation of the t-statistic and influences the resulting p-value for the test of differential expression for a given gene. When the sample size is small, the sample variance can over- or under-estimate the true variance (sometimes wildly), leading on the one hand to overly conservative p-values, or false positives (calling the gene significant due to under-estimating its true variance).

Tools like `limma` or `edgeR` designed for analysing gene expression data employ a clever method to refine the empirical variance estimates for each gene and shrink them towards a common value. It uses the behaviour of the majority of the genes to 'adjust' the poor estimates of others (borrowing information between genes).

Multiple Hypothesis Testing

Classical hypothesis testing results in a p-value indicating the probability that a particular result occurred by chance under the null hypothesis (e.g. that there is no difference in mean expression levels between 2 groups). When the p-value is sufficiently low (<0.05), the null hypothesis is considered unlikely and rejected.

A p-value of 0.05 means a 5% chance of a false positive result (by chance, our small sample shows a difference, even when none really exists). And every time we conduct a hypothesis test, this 5% 'risk' of seeing a significant result by chance exists. Adjusting raw p-values for multiple testing is a way to avoid inflating the type I error rate (false positives) when testing thousands of genes simultaneously.

False discovery rate (FDR) – also known as Benjamini-Hochberg's (BH) method to control the false discovery rate is a common method used in gene expression analysis. Here, the raw p-values are adjusted in a step-wise fashion:

Raw p-value * number of tests/ rank position

Where rank position refers to the raw p-values sorted in ascending order. The smallest raw p-value is penalised most heavily (p*N/1), the second smallest p-value is adjusted to (p*N/2), the third smallest to (p*N/3) and so on. It is widely used in gene expression analysis, since genes are unlikely to be independent of each other (many are correlated), and the Bonferroni correction (setting alpha =0.05/number of tests) assumes independent tests are performed and is too stringent.

In our final inspection of the limma output, we hope to find some genes passing the FDR correction – say there were 300 significant genes with adjusted p-values <0.05. We interpret this as controlling the false discovery rate at 5%: among the set of 300 genes, only 5% of them (15 genes) are expected to be false positives.

It might be there are 1000s of significant genes, or none at all, depending on the power of the experiment (sufficient replicates) and the size of the effect (fold change) and heterogeneity of the samples (human clinical samples will be more variable than mouse models or cell line experiments). Even if no genes pass the multiple testing correction, it is rare for the results to indicate that there are really no changes of interest in the dataset - it is more likely to be an under-powered experiment and therefore the ranked list of genes are usually worth exploring by looking at whether genes of interest come up or pathway analysis for further insight and informing future follow-up work.