Block 5: Hypothesis Tests

Huizhi Lin (881125518130) & Jan Orsel (960608
November 24, 2017

During this project the goal is to find out for which genes the expression levels differ significantly between normal and tumour samples. The genes analysed are the genes in the female reproductive system dataset.

Inspect the data set

We first read the data into a data frame and then inspected the data using the code below.

```
data <-
read.table('get_normal_vs_tumor2_RAW_Female.Reproductive.System.out',
header = TRUE, sep = ' ')
dim(data)

## [1]  130 2562

colnames(data)

##    [1] "NAALAD2"    "NAALADL1"   "ACOT8"      "GNPDA1"     "KCNE3"
##    [6] "GNE"        "HCN4"       "PIGK"       "SLC17A4"    "ABCC5"
##   [11] "ABCB6"      "ABCC9"      "ABCF2"      "ATP9A"      "KCNK7"
##   [16] "UST"        "ADA"        "AASS"       "ATP6AP2"    "LPCAT3"
##   [21] "CHST4"      "SLC25A13"   "SLC25A15"   "DHRS9"      "ALG3"
##   [26] "NME6"       "DHRS2"      "MFSD10"     "COQ7"       "SLC35B1"
##   [31] "KCNMB2"     "GPHN"       "SLC17A2"    "GLYAT"      "ABCC4"
##   [36] "TCIRG1"     "B3GALT5"    "RRAGB"      "AKR1A1"     "B3GNT3"
##   [41] "ABCA7"      "ABCA9"      "ABCA8"      "CACNG3"     "CACNG2"
##   [46] "CDO1"       "BPNT1"      "CEPT1"      "ATP8A1"     "PEMT"
##   [51] "ST3GAL6"    "CDS1"       "CDIPT"      "LYPLA1"     "ACAA2"
etc...
```

This data frame has 130 rows (samples) and 2562 columns (2561 genes and one extra column with the information of tissue samples).

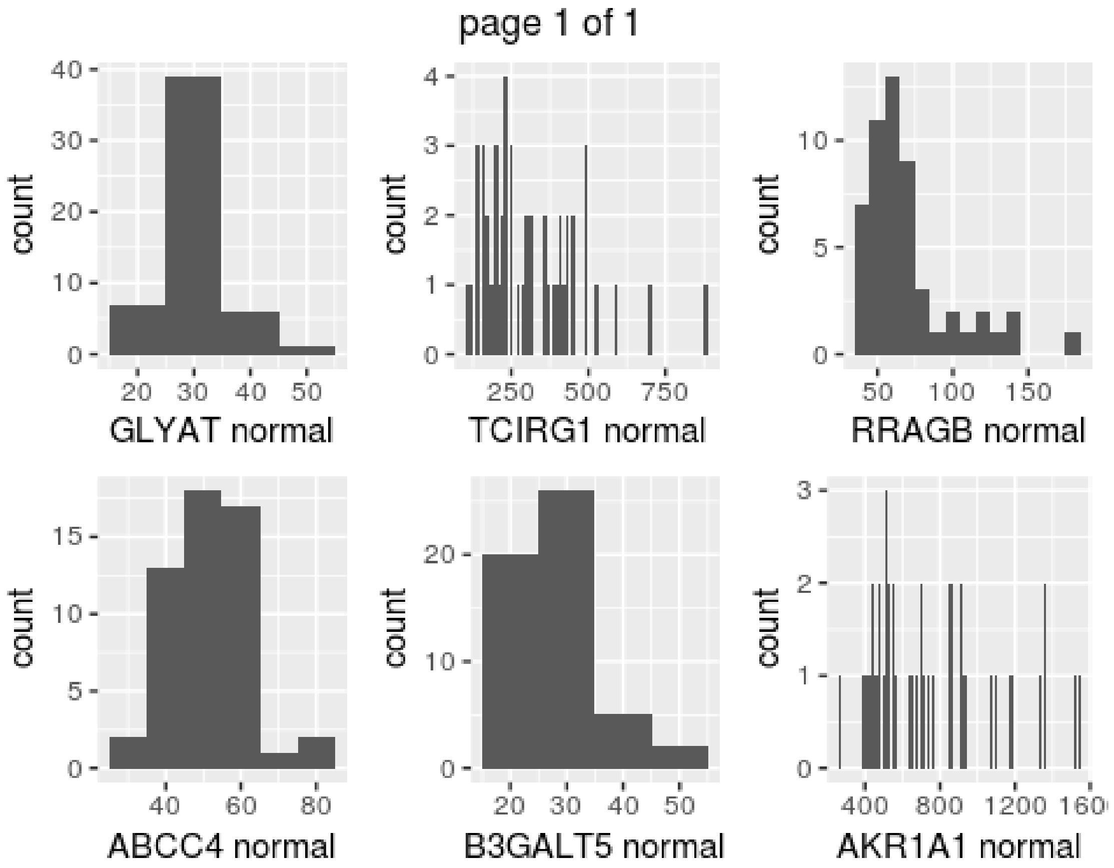
creating dataframes of normal tissue and tumor tissue

Before we could start doing hypothesis testing, we had to create two subdata set, one containing the normal samples and the other one containing the tumour samples.

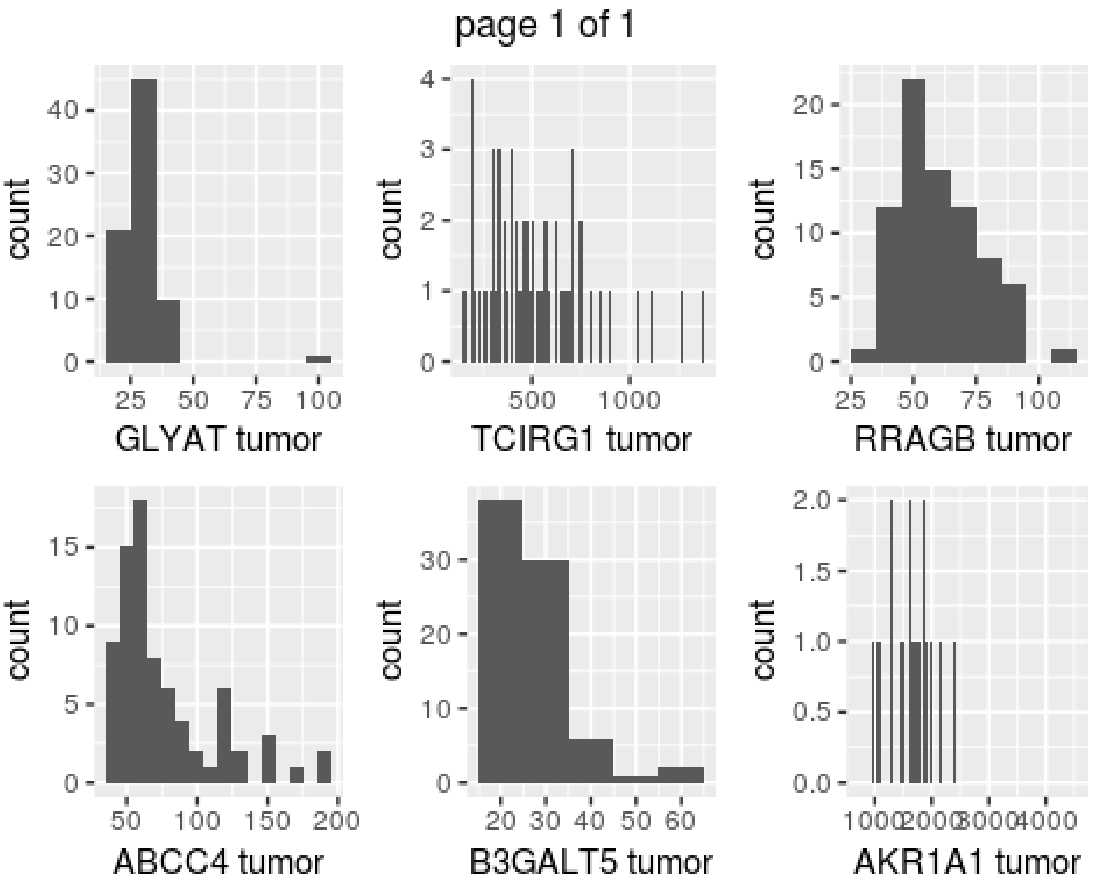
```
normal <- data[data$tissue=='normal',]
tumor <- data[data$tissue=='tumor',]
```

Then we selected 6 genes and created histograms to visualise their distribution. This was done in order to check if the data is normally distributed.

```
library(ggplot2)
library("gridExtra")
pl <- lapply(34:39, function(i){
  ggplot(data=normal,aes(x=normal[,i])) + geom_histogram(binwidth=10) +
xlab(sprintf("%s normal",colnames(normal)[i]))
})
marrangeGrob(pl, nrow=2, ncol=3)
```



```
pl <- lapply(34:39, function(i){
  ggplot(data=tumor,aes(x=tumor[,i])) + geom_histogram(binwidth=10) +
xlab(sprintf("%s tumor",colnames(tumor)[i]))
})
marrangeGrob(pl, nrow=2, ncol=3)
```



The histograms show that the data for most of the genes is not normally distributed, which is futher proved by the *Shapiro-Wilk* normality test shows below. Furthermore it shows that the range of expression varies enormously for each gene. This makes choosing an appropriate bin size harder.

```
# normal samples
for (i in 34:39){
  result <- shapiro.test(normal[,i])
  print(result)
}

##
##  Shapiro-Wilk normality test
##
## data:  normal[, i]
## W = 0.88938, p-value = 0.0001431
##
##
##  Shapiro-Wilk normality test
##
## data:  normal[, i]
## W = 0.97438, p-value = 0.3096
```

```
##
##
##  Shapiro-Wilk normality test
##
## data:  normal[, i]
## W = 0.90924, p-value = 0.0006777
##
##
##  Shapiro-Wilk normality test
##
## data:  normal[, i]
## W = 0.90042, p-value = 0.0003339
##
##
##  Shapiro-Wilk normality test
##
## data:  normal[, i]
## W = 0.81072, p-value = 8.901e-07
##
##
##  Shapiro-Wilk normality test
##
## data:  normal[, i]
## W = 0.91664, p-value = 0.001255
```

```r
# tumor samples
for (i in 34:39){
  result <- shapiro.test(tumor[,i])
  print(result)
}
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tumor[, i]
## W = 0.60773, p-value = 4.965e-13
##
##
##  Shapiro-Wilk normality test
##
## data:  tumor[, i]
## W = 0.83369, p-value = 7.388e-08
##
##
##  Shapiro-Wilk normality test
##
## data:  tumor[, i]
## W = 0.90995, p-value = 4.539e-05
##
##
##  Shapiro-Wilk normality test
```

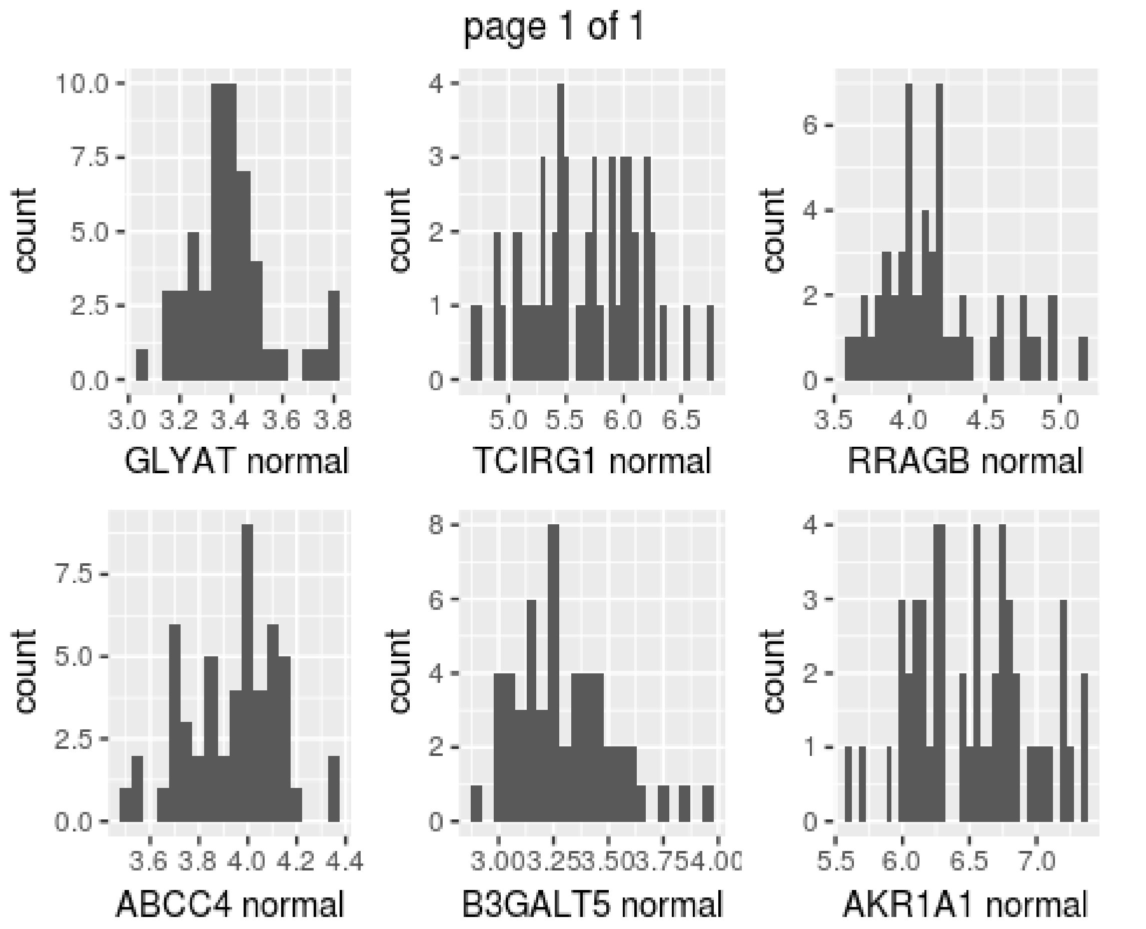
```
##
## data:  tumor[, i]
## W = 0.80724, p-value = 1.216e-08
##
##
##  Shapiro-Wilk normality test
##
## data:  tumor[, i]
## W = 0.94798, p-value = 0.00332
##
##
##  Shapiro-Wilk normality test
##
## data:  tumor[, i]
## W = 0.89976, p-value = 1.674e-05
```

Therefore we decided to create a new data frame with log transformed data in the hope that this data would become normal distributed.

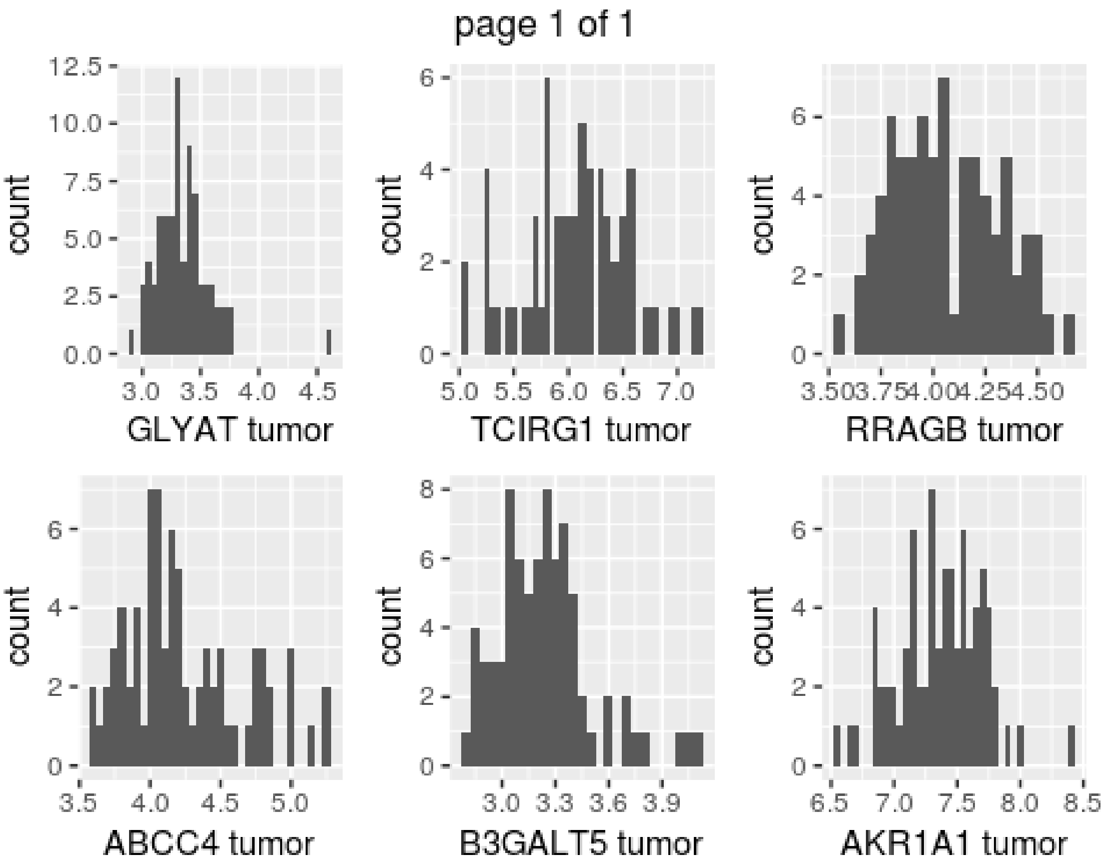
```
logdata <- data
logdata[,1:2561] <- log(logdata[,1:2561])
normal.log <- logdata[logdata$tissue=='normal',]
tumor.log <- logdata[logdata$tissue=='tumor',]
```

After the log transformation, we created histograms of the same 6 genes, and check the distribution with *Shapiro-Wilk* test.

```
pl <- lapply(34:39, function(i){
  ggplot(data=normal.log,aes(x=normal.log[,i])) +
geom_histogram(binwidth=0.05) + xlab(sprintf("%s
normal",colnames(normal.log)[i]))
})
marrangeGrob(pl, nrow=2, ncol=3)
```



```
pl <- lapply(34:39, function(i){
  ggplot(data=tumor.log,aes(x=tumor.log[,i])) +
geom_histogram(binwidth=0.05) + xlab(sprintf("%s
tumor",colnames(tumor.log)[i]))
})
marrangeGrob(pl, nrow=2, ncol=3)
```



```
# normal samples
for (i in 34:39){
  result <- shapiro.test(normal.log[,i])
  print(result)
}

##
##  Shapiro-Wilk normality test
##
## data:  normal.log[, i]
## W = 0.94291, p-value = 0.01348
##
##
##  Shapiro-Wilk normality test
##
## data:  normal.log[, i]
## W = 0.97301, p-value = 0.2713
##
##
##  Shapiro-Wilk normality test
##
## data:  normal.log[, i]
## W = 0.98599, p-value = 0.7866
##
```

```
##
##  Shapiro-Wilk normality test
##
## data:  normal.log[, i]
## W = 0.9656, p-value = 0.1296
##
##
##  Shapiro-Wilk normality test
##
## data:  normal.log[, i]
## W = 0.92662, p-value = 0.002989
##
##
##  Shapiro-Wilk normality test
##
## data:  normal.log[, i]
## W = 0.97474, p-value = 0.3201
```

```
# tumor samples
for (i in 34:39){
  result <- shapiro.test(tumor.log[,i])
  print(result)
}
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tumor.log[, i]
## W = 0.88185, p-value = 3.274e-06
##
##
##  Shapiro-Wilk normality test
##
## data:  tumor.log[, i]
## W = 0.94105, p-value = 0.001406
##
##
##  Shapiro-Wilk normality test
##
## data:  tumor.log[, i]
## W = 0.98683, p-value = 0.6136
##
##
##  Shapiro-Wilk normality test
##
## data:  tumor.log[, i]
## W = 0.92817, p-value = 0.0003133
##
##
##  Shapiro-Wilk normality test
##
```

```
## data:  tumor.log[, i]
## W = 0.97784, p-value = 0.1981
##
##
##  Shapiro-Wilk normality test
##
## data:  tumor.log[, i]
## W = 0.98598, p-value = 0.5603
```

The results shows that after log-transformation, two of the 6 genes in the normal samples and three of the 6 genes in the tumour samples are still not normally distributed.

hypothesis tests

t-test

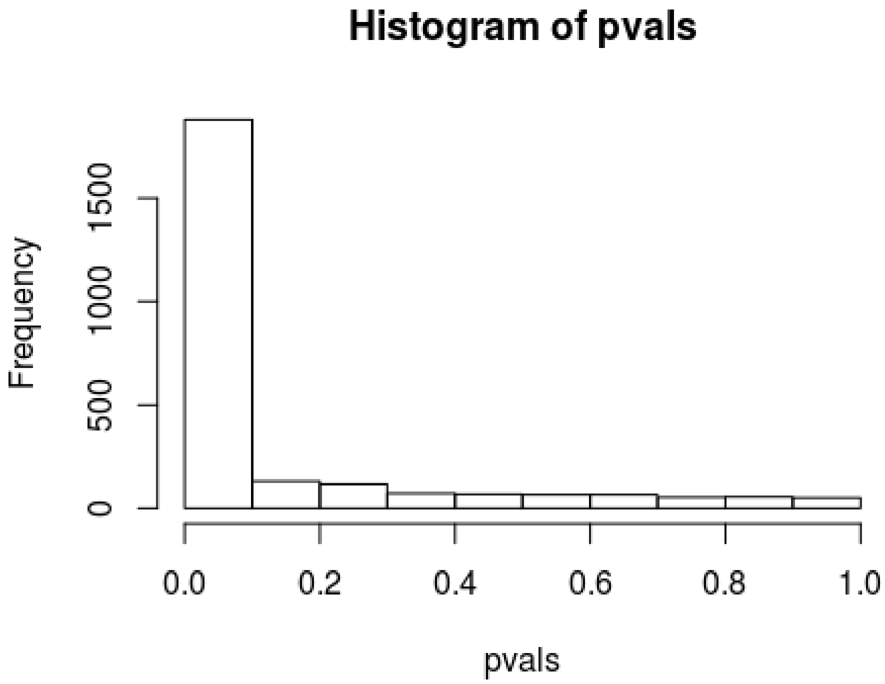
First we chose to use t-test. We decided to use a two sided t test because a gene can either be up-regulated or down-regulated in a normal sample compared to a tumour sample. H0 = There is no difference in expression of the genes analysed between the normal and the tumour samples. Ha = There is a difference in expression of the genes analysed between the normal and the tumour samples.

First we tested only the first gene in the data set to verify that the results are as expected. Then in a for loop all the p-values for all the genes is assigned to a variable called pvals.

```
t.test(normal.log[,1], tumor.log[,1], 'two.sided')

##
##  Welch Two Sample t-test
##
## data:  normal.log[, 1] and tumor.log[, 1]
## t = 9.9496, df = 89.756, p-value = 3.736e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.5952941 0.8923471
## sample estimates:
## mean of x mean of y
##  3.601121  2.857301

pvals <- sapply(1:2561,function(i){
  t.test(normal.log[,i],tumor.log[,i],"two.sided")$p.value
})
hist(pvals)
```



This histogram shows the distribution of p- values, the first bin contains all genes get a p-value under 0.05.

Then we created a new data frame with all the genes name as first column and their p-values as second column.

```
genes <- colnames(logdata)[-2562]
df.ttest <- data.frame(genes,pvals)
df.ttest = df.ttest[order(df.ttest$pvals),]
```

We applied the *Bonferroni* and *Benjamini-Hochberg* methods to limit the probability of false discovery. We also assign the adjust p-values from these two method to the data frame.

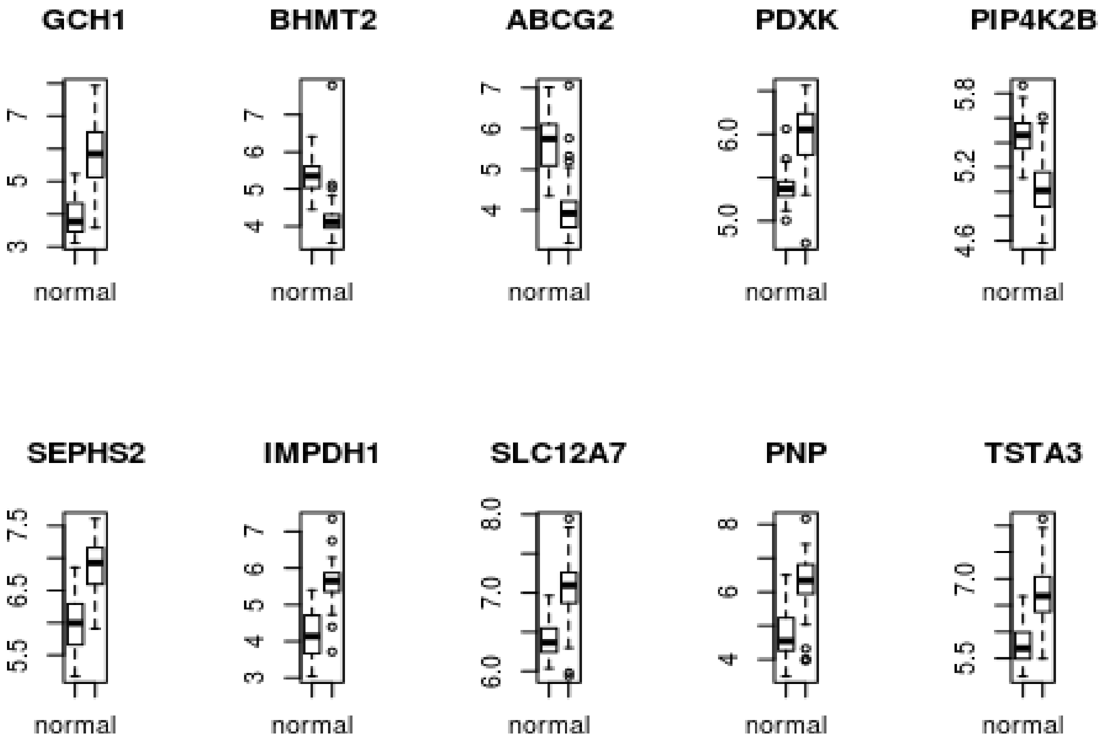
```
df.ttest$Bonferroni <- p.adjust(df.ttest$pvals, method = "bonferroni")
df.ttest$BH <- p.adjust(df.ttest$pvals, method = "BH")
head(df.ttest, n=10)

##        genes        pvals   Bonferroni           BH
## 796     GCH1 8.693206e-30 2.226330e-26 2.226330e-26
## 675    BHMT2 2.038797e-26 5.221359e-23 2.610680e-23
## 2499   ABCG2 2.843381e-25 7.281899e-22 2.427300e-22
## 2347    PDXK 8.011712e-25 2.051799e-21 5.129499e-22
## 2265 PIP4K2B 1.019974e-24 2.612153e-21 5.224305e-22
## 592   SEPHS2 1.305715e-24 3.343936e-21 5.573227e-22
## 1070  IMPDH1 2.773137e-24 7.102004e-21 1.014572e-21
## 85   SLC12A7 6.350533e-24 1.626372e-20 1.851032e-21
## 1302     PNP 6.504995e-24 1.665929e-20 1.851032e-21
## 2051   TSTA3 7.353642e-24 1.883268e-20 1.883268e-21
```

Above here we got our top 10 most significant genes. We made boxplots to check if there are any artefacts. We are looking for very strange looking boxplots where something could have gone wrong with the measuring equipment.

```
par(mfrow=c(2,5))
for (i in 1:10){
  gene <- df.ttest[i,1]
  index <- grep(sprintf("^%s$",gene), colnames(normal.log))

boxplot(normal.log[,index],tumor.log[,index],names=c("normal","tumor"),mai
n=sprintf("%s",gene))
}
```



In our opinion the boxplots do not indicate any clear measuring errors. For some genes the distribution is small, for example the PDXL normal sample is ranges relatively small but there is no evidence that this is incorrect.

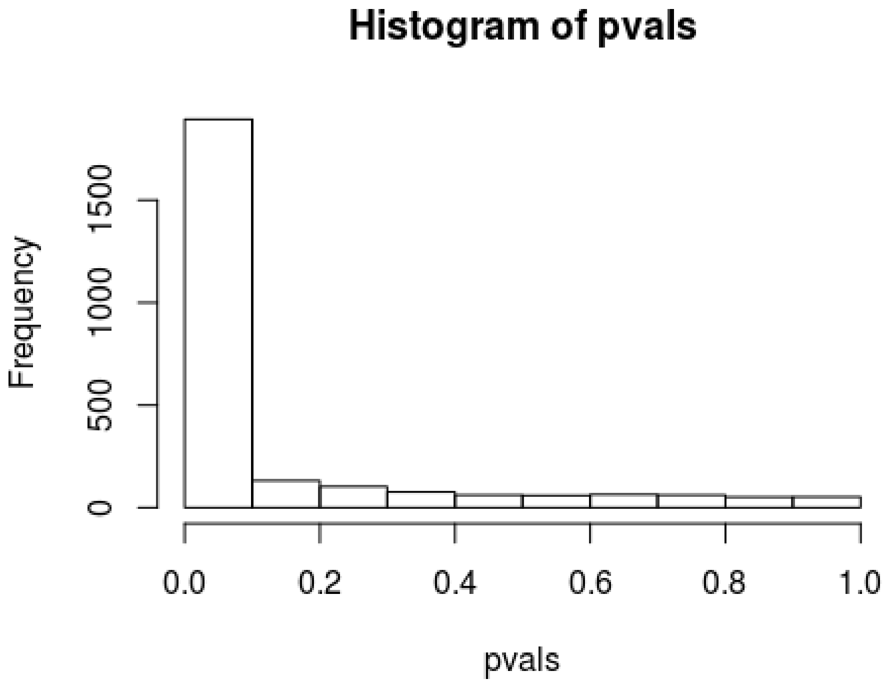
Mann-Whitney U test

Because we found some genes do not follow a normal distribution even after the log-transformation we decided to also do *Mann-Whitney U test*. This test is less powerful but designed to work with data that is not normally distributed..

```
wilcox.test(normal[,1], tumor[,1])

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  normal[, 1] and tumor[, 1]
## W = 3752, p-value = 5.219e-16
## alternative hypothesis: true location shift is not equal to 0

pvals <- sapply(1:2561,function(i){
  wilcox.test(normal[,i],tumor[,i])$p.value
})
hist(pvals)
```



Then we created a new data frame with all the genes name as first column and their p-values as second column. And we applied *Bonferroni* and *Benjamini-Hochberg* method to limit the probability of false discovery.

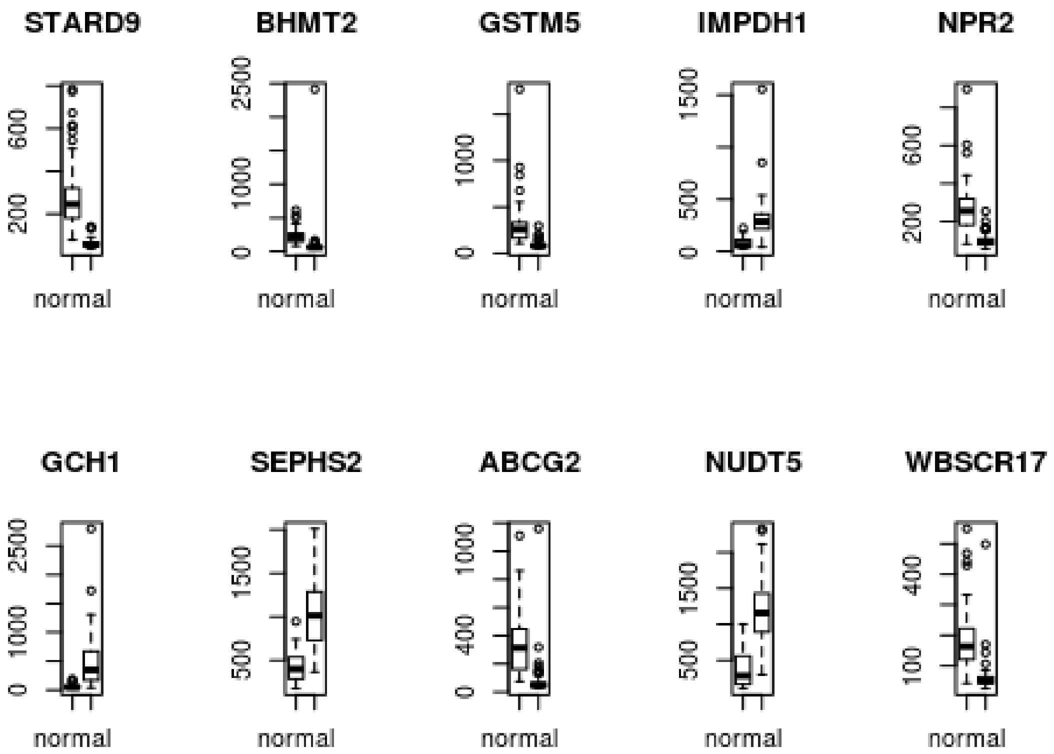
```
genes <- colnames(data)[-2562]
df.utest <- data.frame(genes,pvals)
df.utest = df.utest[order(df.utest$pvals),]
df.utest$Bonferroni <- p.adjust(df.utest$pvals, method = "bonferroni")
df.utest$BH <- p.adjust(df.utest$pvals, method = "BH")
head(df.utest, n=10)
```

```
##        genes        pvals   Bonferroni           BH
## 1765  STARD9 1.072585e-20 2.746889e-17 2.746889e-17
## 675    BHMT2 4.838958e-20 1.239257e-16 6.128313e-17
## 938    GSTM5 7.178813e-20 1.838494e-16 6.128313e-17
## 1070  IMPDH1 1.110649e-19 2.844371e-16 7.110927e-17
## 1306    NPR2 5.473327e-19 1.401719e-15 2.803438e-16
## 796     GCH1 1.035988e-18 2.653166e-15 4.421943e-16
## 592   SEPHS2 1.648738e-18 4.222419e-15 5.988314e-16
## 2499   ABCG2 1.870618e-18 4.790652e-15 5.988314e-16
## 130    NUDT5 2.406531e-18 6.163127e-15 6.701822e-16
## 1879 WBSCR17 2.616877e-18 6.701822e-15 6.701822e-16
```

Here we got our top 10 most significant genes. We made boxplots to check if some of these genes turned out to be significant because of artefacts or machine errors.

```
par(mfrow=c(2,5))
for (i in 1:10){
  gene <- df.utest[i,1]
  index <- grep(sprintf("^%s$",gene), colnames(normal))

boxplot(normal[,index],tumor[,index],names=c("normal","tumor"),main=sprintf("%s",gene))
}
```



The boxplots visualise the distribution of the 10 genes where the difference between normal and tumour samples is most significant. These boxplots are created to check if the difference in significance is derived from actual difference between the two distribution or if the difference is derived form an error in the data. The ten boxplots above these ten don’t show very clearly if that is the case, the ten boxplots derived from the Mann-Whitney U test show some clear candidates where something might have gone wrong. A few plots show a distribution for either normal or tumour tissue that is simply not realistic in real life, especially compared to their tissue counterpart. The boxplot for BHMT2 shows two very small boxes, it could be that this is the natural distribution for this gene but a few very weird outliers suggest otherwise. Some of the genes in this top ten are also in the top ten of the t-test derived data. It would be wise to double check the original data to see if there were any mistakes made by the measuring equipment.