Assessment 5

Joana Levtcheva, CID 01252821

Introduction

The aim of thie report is to reproduce and extend some of the analysis given in the article Microarrays, Empirical Bayes and the Two-Groups Model (Efron, 2008).

We are going to work with microarray prostate cancer data consisting of $N = 6033$ gene expressions for $n_1 = 50$ healthy males and $n_2 = 52$ prostate cancer patients.

Data loading:

```
load("prostate.RData")
```

Data preprocessing:

```
df <- as.data.frame(t(prostatedata))
status_1 <- as.vector(rep(1, 50))
status_2 <- as.vector(rep(2, 52))
df$status <- c(status_1, status_2)
df$status <- as.factor(df$status)
rownames(df) <- NULL
```

Part 1

Performing a two-sample t-test on each of the 6033 genes by extracting the p-values and t-statistics from the model summary of `lm( expression ~ status)`, and storing the t-statistic and p-value in each case. The null hypothesis of the test states that the two status groups, healthy males and prostate cancer patients, have the same mean. This is equivalent to performing two-sample t-test (with the command `t.test` in R with `var.equal` set to `TRUE`) on the two status groups.

```
tvalues <- c()
pvalues <- c()

desired_columns  <- names(df)[!names(df)=="status"]
for (column in desired_columns){
    lm_formula <- formula(paste(column, "~ status"))
    col_lm <- lm(lm_formula, data = df)

    tvalue <- coef(summary(col_lm))[,3][["status2"]]
    tvalues <- append(tvalues, tvalue)

    pvalue <- coef(summary(col_lm))[,4][["status2"]]
    pvalues <- append(pvalues, pvalue)
}
```

Now, let's plot a histogram showing the distribution of the obtained p-values. The histogram is constructed such that the bins have width 0.05, so the interval [0, 0.05] in which the p-values are such that the null hypothesis is rejected can be distinct.

```
pvalues_hist <- hist(
  pvalues,
  breaks = seq(from=0, to=1, by=0.05),
  col = 'skyblue',
  ylim = c(0, 500),
  main = "Histogram of p-values", xlab = "p-values"
)
mtext(
  "Fig. 1."
  ~italic("P-values obtained from performing two-sample t-test on both status groups"),
  side = 1, line = 4, at=0.5, cex=0.75)
```

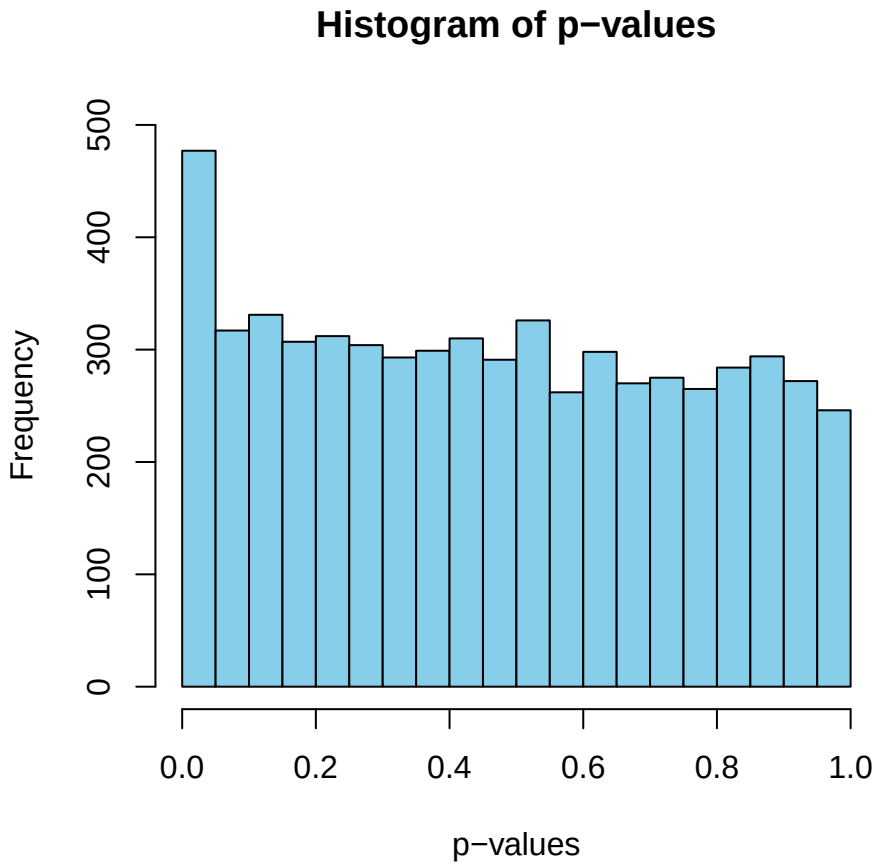

Fig. 1. *P−values obtained from performing two−sample t−test on both status groups*

The distribution of the p-values resembles a uniform distribution, except for the first bin [0, 0.05] which is visibly taller than the rest.This is where the alternative hypothesis lives along with potential false positives. The taller this bin, the more p-values are close to 0 and hence significant, rejecting the null hypothesis that the two groups have equal means. `Fig. 1.` shows low percent of alternative hypothesis.

Under the null hypothesis, the histogram of p-values should look flat and uniformly distributed over the interval [0,1]. This is true because the p-value is the probability integral transform of the test statistic, and the test statistic under the null hypothesis has a t-distribution in the case of performing t-test.

The null p-values are uniformly distributed because as part of the p-value definition we have that under the null hypothesis it has $\alpha\%$ chance of being less than α. In fact, this describes a uniform distribution.

We are going to empirically demonstrate that the p-values are uniformly distributed by randomly permutating the status variable. By doing this we are destroying all group information.

We are permutating only the status variable and repeating the analysis:

```
set.seed(123)
df_permutated = transform(df, status = sample(status))

tvalues_permutated <- c()
pvalues_permutated <- c()
desired_columns  <- names(df_permutated)[!names(df_permutated)=="status"]
for (column in desired_columns){
    lm_formula <- formula(paste(column, "~ status"))
    col_lm <- lm(lm_formula, data = df_permutated)

    tvalue <- coef(summary(col_lm))[,3][["status2"]]
    tvalues_permutated <- append(tvalues_permutated, tvalue)

    pvalue <- coef(summary(col_lm))[,4][["status2"]]
    pvalues_permutated <- append(pvalues_permutated, pvalue)
}
```

```
pvalues_permutated_hist <- hist(
  pvalues_permutated,
  breaks = seq(from=0, to=1, by=0.05),
  col='skyblue',
  ylim = c(0, 500),
  main = "Histogram of p-values", xlab = "p-values"
)
mtext(
  "Fig. 2."
  ~italic("P-values from performing two-sample t-test with permutated status variable"),
  side = 1, line = 4, at=0.5, cex=0.75
)

xfit <- seq(min(pvalues_permutated), max(pvalues_permutated), length = 2)
yfit <- dunif(xfit, min = min(pvalues_permutated), max = max(pvalues_permutated))
yfit <- yfit * diff(pvalues_permutated_hist$mids[1:2]) * length(pvalues_permutated)
lines(xfit, yfit, col = "red", lwd = 2)
legend(0.4, 475, legend=c("Uniform distribution"), lty=1, col=c("red"), box.lty=0)
```

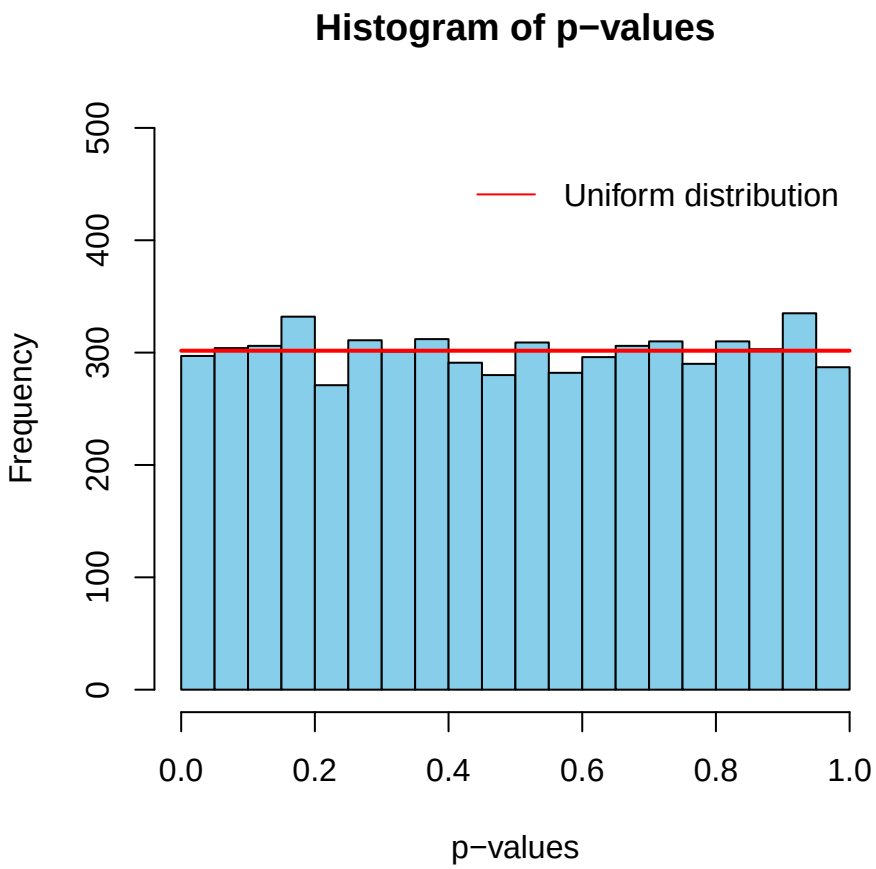

Fig. 2. *P−values from performing two−sample t−test with permutated status variable*

The distribution in `Fig. 2.` looks like an uniform distribution. This resembles the case of what p-values would look like if all null hypothesis had been true. This tells us that not all of them are null but rather at most a small percentage of the hypothesis are not true. Later, a FDR (False Discovery Rate) method will be used to identify these cases.

We are going to compare our empirical null distribution of the p-values obtained from the t-tests performed after permutating the status variable with its theoretical expectation (the uniform distribution) by using a p-p plot.

```
probDist <- punif(pvalues_permutated)
plot(
  ppoints(length(pvalues_permutated)),
  sort(probDist),
  main = "PP Plot",
  xlab = "Empirical Null Distribution",
  ylab = "Uniform Distribution"
)
abline(0,1)
mtext(
  "Fig. 3."
  ~italic("P-P plot comparing the empirical null distribution with Uniform distribution"),
  side = 1, line = 4, at=0.5, cex=0.75
)
```

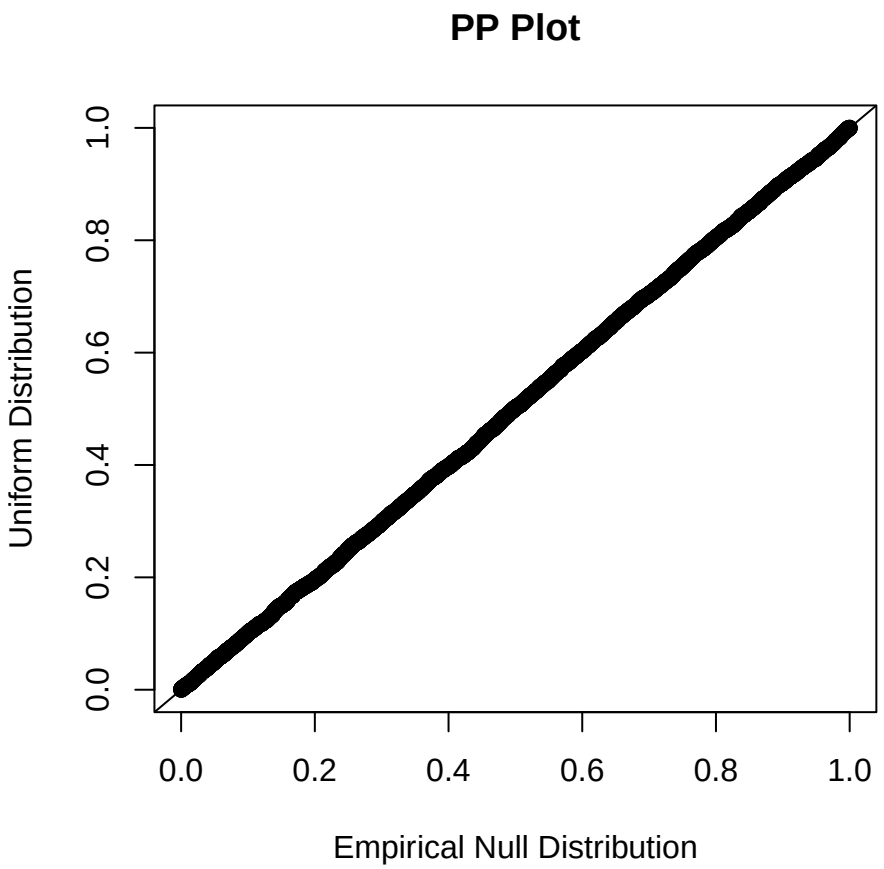

Fig. 3. *P–P plot comparing the empirical null distribution with Uniform distribution*

A p-p plot visualizes the CDFs of two distributions (the empirical and theoretical distribution) against each other. To compare the distributions, we check if the points lie on a 45 degree line from (0,0) to (1,1). If they do not deviate the distributions do not differ. The p-p plot in Fig. 3. shows that after randomly permutating the status variable, the points representing the empirical null distribution follow the straight line and there are no deviations from it. Therefore the empirical null distribution matches the theoretical uniform distribution.

Part 2

Equation 3.2 of the article states $z_i = \Phi^{-1}(F_{100}(t_i))$, where Φ indicates the standard normal CDF and F_{100} is the CDF of a standard t-distribution with 100 df, and t_i are the two-sample t-statistic for every gene comparing the healthy males versus the prostate cancer patients. The formula is used to transform these t-values to z-values. Theoretically the z_i value should have a standard normal distribution $N(0,1)$ if gene i produces identically distributed normal gene expressions for healthy males and prostate cancer patients.

Transforming the t-values to z-values and creating a histogram of the obtained z-values with exactly 49 bins with width 0.2:

```
zvalues <- qnorm(pt(tvalues, df = 100))

zvalues_hist <- hist(
  zvalues,
  breaks = seq(from=min(zvalues), to=max(zvalues)+0.2, by=0.2),
```

```
  col = "skyblue", main="Histogram of z-values", xlab = "z-values", ylim=c(0,500)
  )

xfit <- seq(min(zvalues), max(zvalues), length = 100)
yfit <- dnorm(xfit, mean = 0, sd = 1)
yfit <- yfit * diff(zvalues_hist$mids[1:2]) * length(zvalues)

lines(xfit, yfit, col = "red", lwd = 2)
legend(2, 450, legend=c("N(0,1)"), lty=1, col=c("red"), box.lty=0)
mtext(
  "Fig. 4."
  ~italic("Histogram of the N = 6033 z-values and the theoretical null N(0,1)"),
  side = 1, line = 4, at=c(0), cex=0.75
)
```

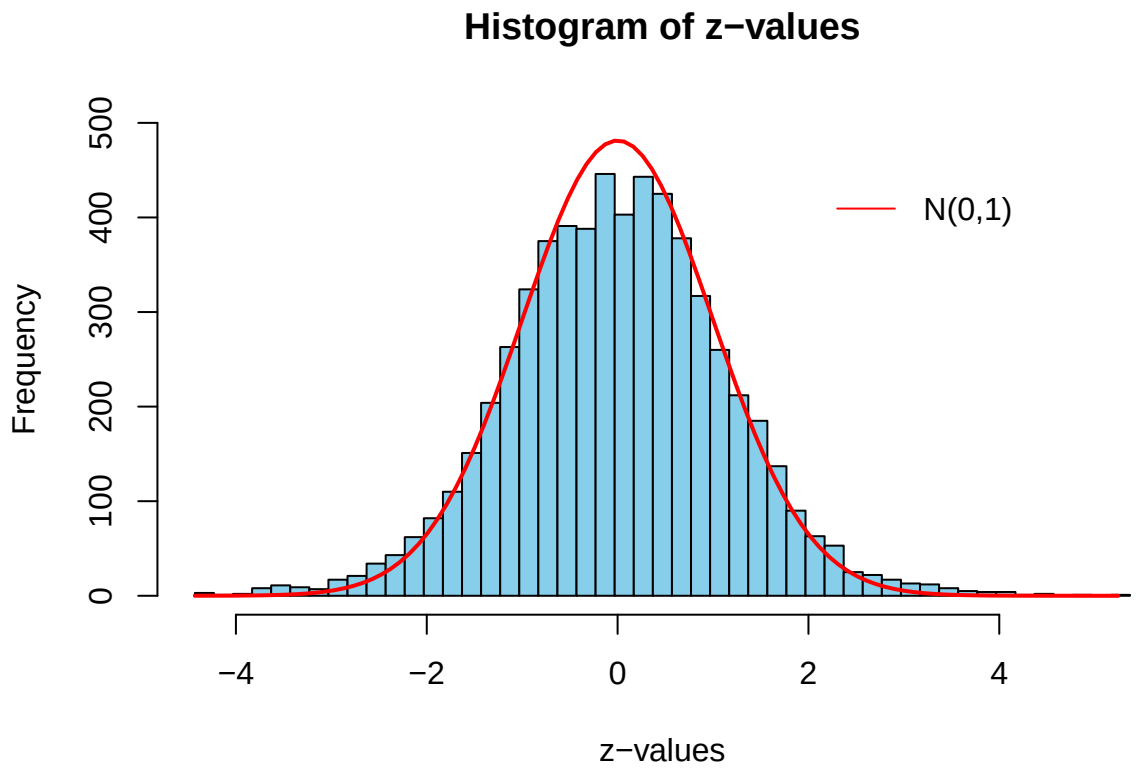

Fig. 4. *Histogram of the N = 6033 z−values and the theoretical null N(0,1)*

The histogram of the z-values in `Fig. 4.` has a normal-shaped central peak which would probably represent the majority of null genes, the ones behaving similarly for healthy males and prostate cancer sufferers, whereas the interesting (non-null) genese would be revealed in the tails.

Part 3

Let's y_k denote the number of genes with effect size in bin k of the histogram in Fig. 4., and also fit Poisson GLMs to estimate the parameters of models of the form $f(z) = e^{\sum_{j=0}^{p} \beta_j x^j}$ (Equation 3.6 from the article), with different values of the polynomial degree p: $p \in \{2, ..., 8\}$.

```r
y <- zvalues_hist$counts
midpoints <- zvalues_hist$mids
aic_values <- c()

for (i in c(2:8)) {
  pol <- poly(midpoints, i, raw = FALSE)
  res <- glm(
    formula = y ~ pol,
    family = poisson(link = "log"),
  )
  aic_values <- append(aic_values, res$aic)
}
```

```r
plot(aic_values, pch = 16, xlab = "p", xaxt = "n")
axis(side = 1, at = 1:7, labels = c(2:8))
mtext(
  "Fig. 5."
  ~italic("AIC values from fitting Poisson GLMs with polynomials of degree p"),
  side = 1, line = 4, at=c(4), cex=0.75
)
```

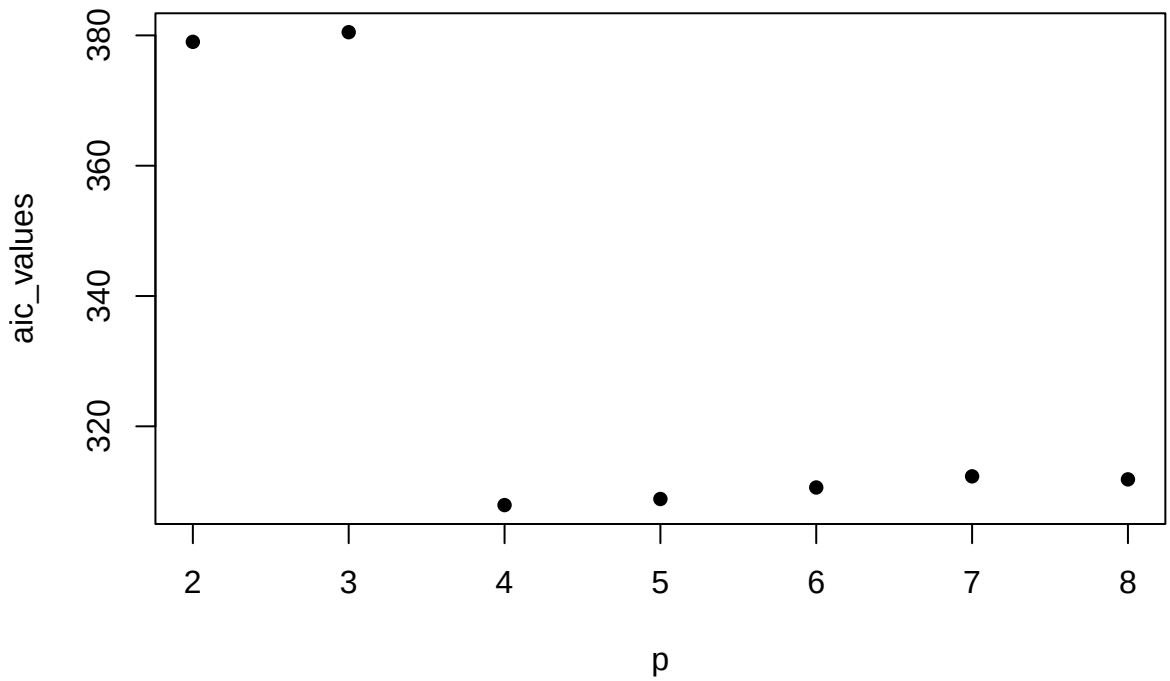

Fig. 5. *AIC values from fitting Poisson GLMs with polynomials of degree p*

Based on the AIC results from `Fig. 5.` the best AIC, corresponding to the lowest AIC, is for degree $p = 4$.

Let's fit a polynomial of degree 4 constructed from the midpoints of each bin of the histogram in Fig. 4. We aim to model the $log(\nu_k)$ with this polynomial. The expectation of y_k: $E(y_k) = \nu_k$ is proportional to the mixture density $f(z)$. By fitting the polynomial of degree 4 via Poisson GLM with log-link we are going to

estimate the coefficients β_j in $f(z) = e^{\sum_{j=0}^{p} \beta_j x^j}$.

```
pol <-poly(midpoints, degree = 4, raw = TRUE)
glm_4 <- glm(
    formula = y ~ pol,
    family = poisson(link = "log"),
)
summary(glm_4)
```

```
##
## Call:
## glm(formula = y ~ pol, family = poisson(link = "log"))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.17001  -0.59298  -0.00221   0.58087   1.80119
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  6.1068421  0.0167798 363.941   <2e-16 ***
## pol1         0.0079974  0.0165227   0.484    0.628
## pol2        -0.4726005  0.0116927 -40.418   <2e-16 ***
## pol3        -0.0025388  0.0026263  -0.967    0.334
## pol4         0.0096637  0.0009294  10.397   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 8989.596  on 48  degrees of freedom
## Residual deviance:   43.033  on 44  degrees of freedom
## AIC: 307.91
##
## Number of Fisher Scoring iterations: 5
```

We can get the estimated coefficients β_j, $j = 1, ..., 4$ from the model summary

```
fz <- function(x) {
  return(exp(6.1068421 + 0.0079974 * x - 0.4726005 * x^2 - 0.0025388  * x^3 + 0.0096637 * x^4))
}
```

and plot $f(z)$ against the histogram of the z-values and the null distribution N(0,1) from Fig. 4:

```
zvalues <- qnorm(pt(tvalues, df = 100))
zvalues_hist <- hist(
  zvalues,
  breaks = seq(from=min(zvalues), to=max(zvalues)+0.2, by=0.2),
  col = "skyblue", main="Histogram of z-values", xlab = "z-values", ylim = c(0, 500)
)

xfit <- seq(min(zvalues), max(zvalues), length = 100)
yfit <- dnorm(xfit, mean = 0, sd = 1)
```

```r
yfit <- yfit * diff(zvalues_hist$mids[1:2]) * length(zvalues)

lines(xfit, yfit, col = "red", lwd = 2)

yfitN <- fz(xfit)
lines(xfit, yfitN, lwd = 2, col = "black")

legend(2, 450, legend=c("N(0,1)", "f(z)"), lty=1, col=c("red", "black"), box.lty=0)
mtext(
  "Fig. 6."
  ~italic("Histogram of the N z-values, theoretical null N(0,1) and estimated mixture denisty f(z)"),
  side = 1, line = 4, at=c(0), cex=0.75
)
```

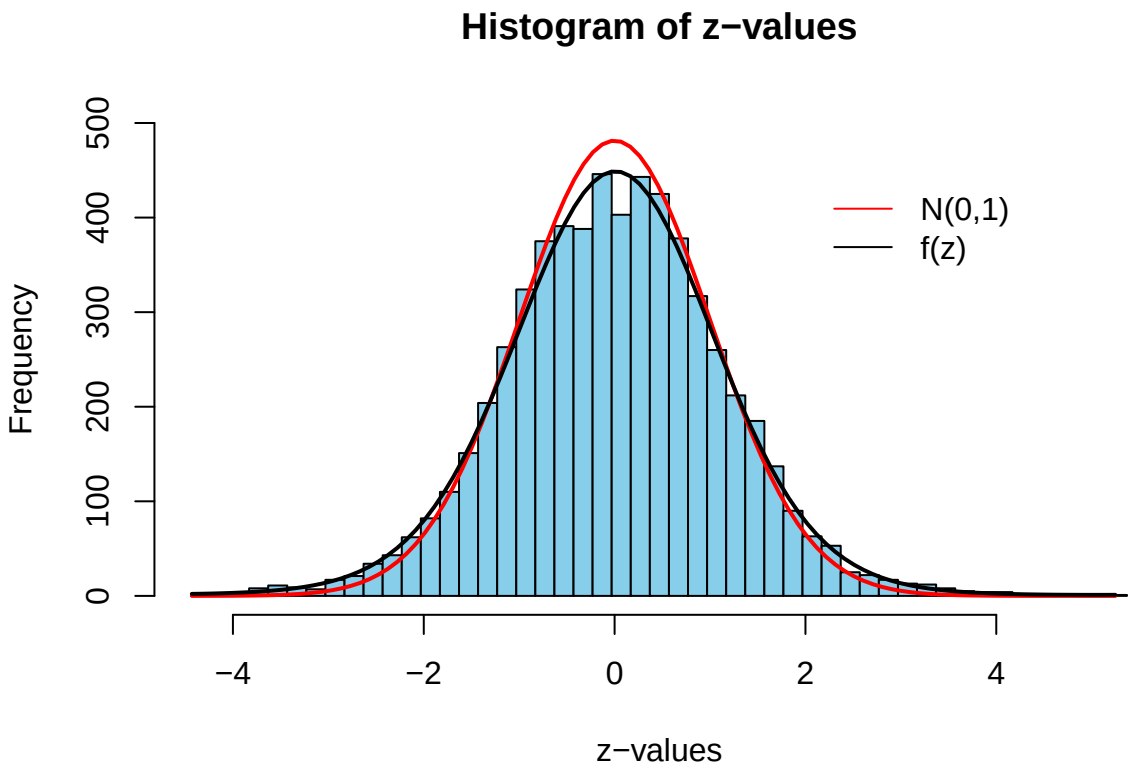

Fig. 6. *Histogram of the N z−values, theoretical null N(0,1) and estimated mixture denisty f(z)*

The histogram in `Fig. 6.` shows that the estimated mixture density $f(z)$ seems to have heavier tails and lower peak compared to the standard normal distribution $N(0,1)$.

Part 4

The local fdr $\hat{fdr}$ can be estimated by the following equation $\hat{fdr} = p_0 * f_0(z)/\hat{f}(z)$, where $p_0 = 0.93$ by problem statement, $f_0(z)$ is the density of null distribution, or $N(0,1)$ in our case, and $\hat{f}(z)$ is the polynomial with coefficients estimated from Part 3.

In Part 3 we directly fitted a polynomial of degree $p = 4$ which estimated its coefficients β_j. But, as mentioned above, the expectation of the y_k: $E(y_k) = \nu_k = N\Delta f(x_k)$ is nearly proportional to the mixture

density f(z), where x_k is the midpoint of interval from the histogram in `Fig. 4.`. We were aiming at modelling $log(\nu_k) = log(N\Delta f(x_k)) = log(N) + log(\Delta) + log(f(x_k))$. But in the R code the model was run with estimating y with a polynomial of the midpoints of degree 4. So, we should scale the estimation of f(z) to correspond to one bin ($\hat{f}(z)$): y_k to be modelled by $N\Delta\hat{f}(x_k)$.

If we take a look at $log(\nu_k) = log(N) + log(\Delta) + log(f(x_k))$ and the fitted polynomial $e^{\sum_{j=0}^{p}\beta_j x^j}$, then we have $log(N) + log(\Delta) + log(f(x_k)) = e^{\sum_{j=0}^{p}\beta_j x^j}$. Or, $log(f(x_k)) = \sum_{j=0}^{p}\beta_j x^j - log(N) - log(\Delta)$.

Finally,

$$\hat{f}(z) = e^{(\beta_0 - log(N) - log(\Delta)) + \beta_1 z + \beta_2 z^2 + \beta_3 z^3 + \beta_4 z^4} = \frac{e^{\beta_0 + \beta_1 z + \beta_2 z^2 + \beta_3 z^3 + \beta_4 z^4}}{N\Delta}$$

.

Therefore, when calculating the local fdr we should take into account this scaling, and for example divide the estimated $f(z)$ by N and Δ so we can get the desired $\hat{f}(z)$ used in the local fdr equation.

We can define the local fdr as follows:

```
p0 <- 0.93
local_fdr <- function(x) {
  return(6033 * 0.2 * p0 * dnorm(x, mean = 0, sd = 1) / fz(x))
}
curve(
  local_fdr(x),
  xlim = c(min(zvalues), max(zvalues)),
  main="Estimated local fdr", xlab="z-value"
)
mtext(
  "Fig. 7."
  ~italic("Estimated local fdr curve, 26 genes on the right and 27 on the left have fdr(zi) < 0.20"),
  side = 1, line = 4, at=c(0), cex=0.75
)
```

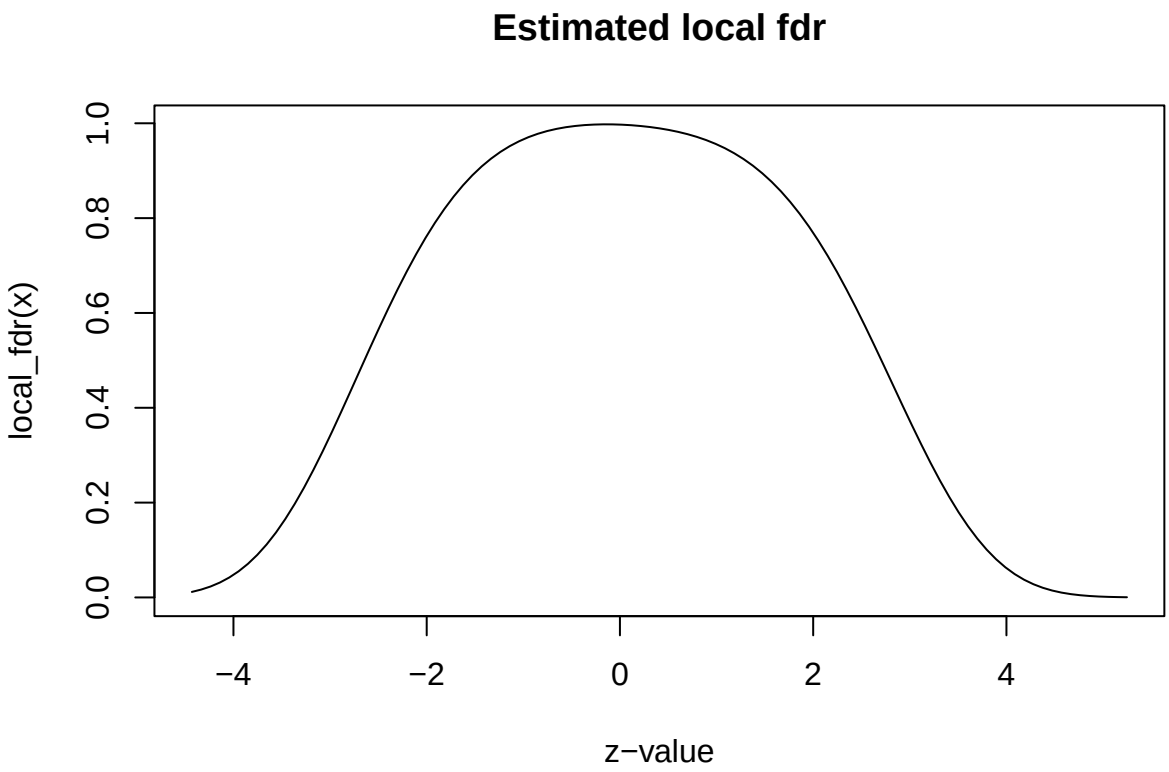

Fig. 7. *Estimated local fdr curve, 26 genes on the right and 27 on the left have fdr(zi) < 0.20*

In order to estimate the number of genes in bins with local fdr < 0.2, we should first find for which z-values fdr < 0.2. The local fdr is approximately 0.2 for z-value = -3.37 (gives fdr ~ 0.2) and z-value = 3.43 (gives fdr ~ 0.2)

For the left part of the local fdr, corresponding to z-value = -3.37, we get 27 interesting genes:

```
length(zvalues[zvalues < -3.37])
```

```
## [1] 27
```

whereas for the right part, corresponding to z-value = 3.43, we get 26 interesting genes.

```
length(zvalues[zvalues > 3.43])
```

```
## [1] 26
```

In `Fig. 7.` the heavy curve is estimated local false discovery rate $\hat{fdr}(z)$ for prostate data. The conditional probability of a null gene given z, declines from 1 when z-value is around 0 to 0 in the opposite ends. There are 53 genes, 26 on the right and 27 on the left, havving $fdr(z_i) < 0.20$. These genes can be considered interesting candidates for further study.

Part 5

The zero assumption 4.1 in the article, states "Most of the z-values near 0 come from null genes". If f_0 is normal $f(z)$ should be well-approximated near $z = 0$ by $p_0\phi_{\sigma_0\delta_0}(z)$ (refer to Equation 4.2 from the article).

We are going to use Equation 4.3 from the article which concerns the $log(f(z)) = \sum_{j=0}^{p} \beta_j x^j = 6.1068421 - log(6033) - log(0.2) + 0.0079974 * z - 0.4726005 * z^2 - 0.0025388 * z^3 + 0.0096637 * z^4$ (again taking into account the scaling constants N and Δ).

The quadratic approximation is actually the second-order Taylor polynomial at point 0,

$$f(z) \approx f(0) + f'(0)z + f''(0)z^2 = \beta_0 + \beta_1 z + \beta_2 z^2 =$$

$$= 6.1068421 - log(6033) - log(0.2) + 0.0079974 * z - 0.4726005 * z^2$$

. Therefore, it is enough to match the previously obtained $\beta_0, \beta_1, \beta_2$ coefficients of $\hat{f}(z)$ with the ones from Equation 4.3 in the article. We have:

```
sigma <- sqrt(1 / (2 * 0.4726005))
sigma
```

```
## [1] 1.02858
```

```
delta<- 0.0079974 * (sigma ^ 2)
delta
```

```
## [1] 0.008461057
```

```
log_p0 <- (
  6.1068421 - log(6033)- log(0.2)) + 0.5 * (((delta^2) + (sigma^2)*log(2*pi*(sigma^2))) / (sigma ^ 2)
)
log_p0
```

```
## [1] -0.04156846
```

Finally, for p_0 we get $p_0 \approx 0.96$:

```
p_0 <- exp(log_p0)
p_0
```

```
## [1] 0.9592837
```

The values obtained for $\sigma_0 = 1.02858$ and $\delta_0 = 0.008461057$ seem logical. If we refer to the histogram in `Fig. 6.`, the estimated f(z) resembles really well the N(0,1) distribution. There is no visible skew compared to the theoretical distribution $N(0,1)$, so the `delta_0` being close to 0 is reasonable. As for the σ_0, there is almost no difference in terms of kurtosis compared to $N(0,1)$, so again having `sigma` close 1 is what was expected.

Part 6

Conclusion Multiple testing consists of having N null hypothesis to consider at the same time, each having its own test statistic and their p-values. Microarray data is such that it may contain thousounds of gene expressions levels which leads to performing thousounds of hypothesis tests simultaneously. For each test there is a possibility fot two types of errors Type I error (fallse positive), meaning a gene is declared to be differentially expressed when it is not, or Type II error (false negative), when the test fails to identify a truly differentially expressed gene. Logically, when the number of simultaneous tests becomes larger, the

probability of Type I or Type II errors (or getting a significant result due to chance) gets higher. Moreover, there might also be computational issues due the high amount of data and needed computational power.

In the context of the current case study, large-scale testing aims to identify a small percentage of interesting cases, gene expressions, that can be further studied. The false discovery rate (FDR) is a simultaneous testing approach which aims at determining the proportion of false positives among all significant results.

In order to estimate the fdr we first estimated the z-values mixture density $f(z)$, denoting it as $\hat{f}(z)$, where the z-values are obtained from the test statistics from performing multiple two-sample t-tests on all of the gene expressions, testing wether the means of the two groups, healthy males versus prostate cancer patients, are equal. The estimation of $f(z)$ is based on dividing the z-values into 49 bins with width 0.2, taking the counts of the z-values in each bin. Because they are following a Poisson distribution this allows for fitting Poisson GLMs, more specficially modelling the logarithm of the z-values counts as a polynomial of degree p. Doing multiple fits for $p \in 2, ..., 8$ lead to chosing degree $p = 4$ as the most suitable polynomial degree based on having the lowest AIC result, and fitting the Poisson GLM produced its coefficient estimates.

After that, the fdr can be estimated via the formula $\hat{fdr}(z) = p_0 f_0 z / \hat{f}(z)$, where $f_0(z)$ is the null distribution density, assumed to be the standard normal distribution $N(0, 1)$, and p_0 is the null prior probability. The estimation of fdr in the current analysis was performed with $p_0 = 0.93$. We are defining a case to be interesting if it has $\hat{fdr}(z) \leq 0.2$, and based on the estimation 53 of the 6033 genese have $\hat{fdr}(z) \leq 0.2$, 26 on the right and 27 on the left. These could be reported to the investigators as likely nonnull candidates (interesting cases) for further experiemntal study.

Literature

- Efron, Bradley., Microarrays, Empirical Bayes and the Two-Groups Model
- Efron, Bradley., Large-Scale Simultaneous Hypothesis Testing: The Choice of a Null Hypothesis
- Efron, Bradley., Local False Discovery rates
- Murdoch, Duncan J., et al., p-Values are Random Variables