Practical 2: The epigenetic clock: Variable ranking and multiple testing

Verena Zuber, Jelena Besevic, Alpha Forna, and Saredo Said

31/1/2019

Part 1: The epigenetic clock: Epigenetic marks associated with ageing

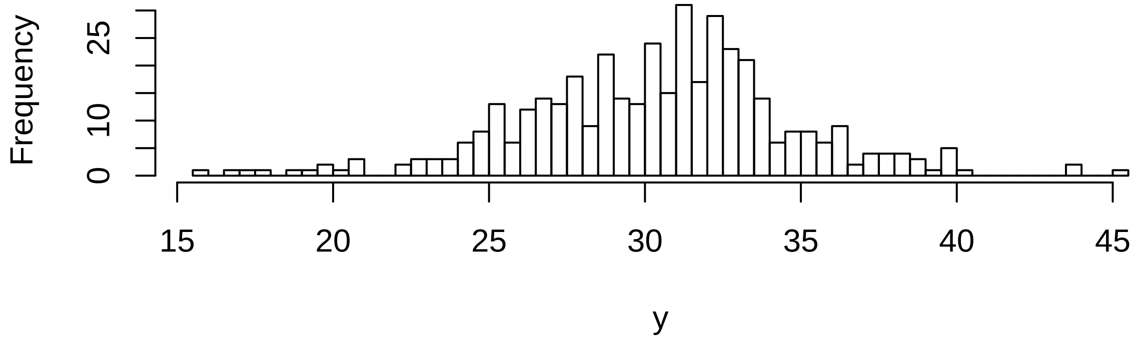
Our epigenome is highly impacted by environmental changes. One particular interesting aspect is ageing and how epigentic marks such as methylation are affected by ageing. Scientists have shown that there exist specific methylation sites that correlate with age, an observation that has been made in humans, chimpanzees, mice, or rats. Based on our knowledge which methylation sites correlate with healthy ageing we can use these epigenetic marks as biomarkers to predict the "actual biological age" of an individual. For example, if an individual has been exposed to pollutants or suffered from stress, its "actual biological age" of its body might be much older than its true age.

If you want to read more on the topic, there is a Nature Feature on the scientist Steve Horvath who first proposed to use methylation to measure the biological age

https://www.nature.com/news/biomarkers-and-ageing-the-clock-watcher-1.15014

In this practical we consider data on $n = 409$ healthy mice and methylation of $p = 3,663$ conserved methylation sites. Load the dataset, that contains the methylation matrix as predictor matrix and the age of the mice (in months) stored in the vector y. Familiarise yourself with the dataset using the following commands

```r
load("data_epigenetic_clock_control")
#alternatively try load("data_epigenetic_clock_control.dms")
y = control_mice$y_control
hist(y,breaks=50, main="")
```



```r
x = control_mice$x_control
dim(x)
```

```
## [1]  409 3663
```

The first part is concerned with performing a ranking of methylation sites that have the strongest association with ageing.

Question 1.1

Compute a linear regression of the first methylation site against the age of the mice. Note in order to access the first column of a matrix, use square bracket like this [,1] and for the jth variable use [,j]. Figure out which element in the $coefficients matrix contains the p-value of the regression coefficient. Use again the squared brackets to index only the $p-$value.

Reply: We fit the regression using the lm() function and look at the value $coefficients.

```
summary(lm(y~x[,1]))$coefficients
```

```
##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 35.673054   2.000145 17.835231 5.451132e-53
## x[, 1]      -6.569309   2.498731 -2.629058 8.886444e-03
```

The element in the 2nd row and the 4th columns of the $ coefficient value contains the $p-$value.

```
summary(lm(y~x[,1]))$coefficients[2,4]
```

```
## [1] 0.008886444
```

Question 1.2

In order to compute the massively univariate linear regression estimate, we need to automate this computation for all $p = 3,663$ methylation sites. First initiate a vector where to save your $p-$values.

```
pvec = rep(NaN, 3663)
```

Write a 'for loop' to iterate through all variables. In case you are not familiar with the 'for loop', use this practical here for help https://www.r-bloggers.com/how-to-write-the-first-for-loop-in-r/ . In each iteration j save the $p-$value of the respective regression into the pvec vector at position pvec[j].

Reply: This is how such a for loop could look like.

```
for(j in 1:ncol(x)){
    pvec[j]=summary(lm(y~x[,j]))$coefficients[2,4]
}
```

Question 1.3

Rank the methylation sites according to their $p-$values and show the top 10 methylation sites that are associated with ageing.

Reply: First we combine the methylation names with the $p-$values and than we sort the output using the order function.

```
output=cbind(colnames(x),pvec)
output_sorted=output[order(pvec),]
head(output_sorted,n=10)
```

```
##                          pvec
##  [1,] "chr13_43122804" "2.22638293312029e-10"
##  [2,] "chr7_118434508" "2.83341285695823e-08"
##  [3,] "chr8_125759184" "8.08770585470011e-08"
##  [4,] "chr7_62419850"  "1.54558113496926e-07"
##  [5,] "chr7_99238006"  "5.02013794094029e-07"
##  [6,] "chr14_78777456" "1.20194579058332e-06"
##  [7,] "chr12_82783446" "1.2180090770785e-05"
##  [8,] "chr15_60172240" "1.71172214464869e-05"
##  [9,] "chr2_72217019"  "2.100193513036e-05"
## [10,] "chr6_121015046" "4.01317786827999e-05"
```

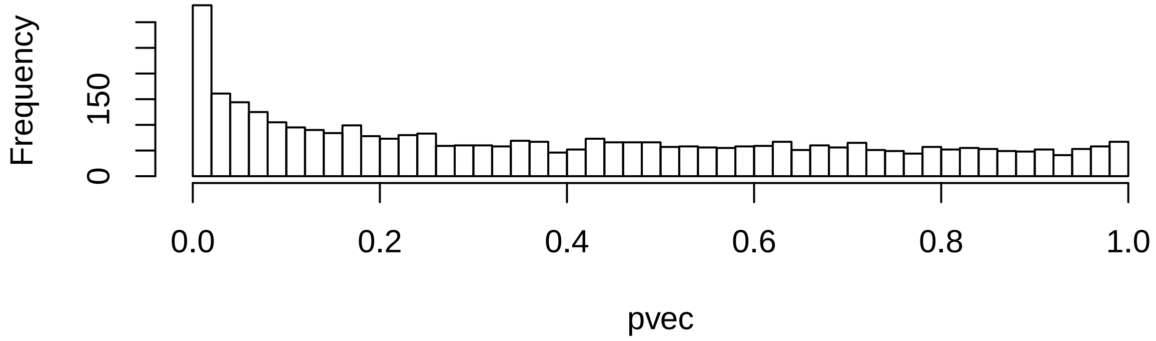
Part 2: FDR control of the variable ranking

The next step is to correct the p-values for multiple testing. Since we performed $p = 3,663$ tests we need to make sure we do control the rate of false positive discoveries. There are different approaches to correct for multiple testing. Here, we perform the key methods and discuss the results in the end.

Question 2.1 Plot a histogram and discuss which distribution this vector of $p-$values has.

Reply: We see here a mixture distribution of a uniform distribution which represents the Null variables and a steep peak at around zero which contains the signal variables.

```
hist(pvec, breaks = 50, main="")
```



Question 2.2 Perform the Bonferroni correction on your $p-$value vector. Use $\alpha = 0.05$ as your significance threshold. How many methylation sites would be considered as Non-Null after Bonferroni correction?

Reply: Using Bonferroni correction we detect 7 methylation sites to be significant.

```
bonferroni = p.adjust(pvec, method="bonferroni")
table(bonferroni<0.05)
```

```
##
## FALSE  TRUE
##  3656     7
```

```
output[which(bonferroni<0.05),]
```

```
##                        pvec
## [1,] "chr12_82783446" "1.2180090770785e-05"
## [2,] "chr13_43122804" "2.22638293312029e-10"
## [3,] "chr14_78777456" "1.20194579058332e-06"
## [4,] "chr7_62419850"  "1.54558113496926e-07"
## [5,] "chr7_99238006"  "5.02013794094029e-07"
## [6,] "chr7_118434508" "2.83341285695823e-08"
## [7,] "chr8_125759184" "8.08770585470011e-08"
```

Question 2.3 Perform the Benjamini-Hochberg Fdr correction on your $p-$value vector. Use $\alpha = 0.05$ as your significance threshold. How many methylation sites would be considered as Non-Null after Benjamini-Hochberg Fdr correction?

Reply: Using Benjamini Hochberg correction we detect 35 methylation sites to be significant.

```
bh = p.adjust(pvec, method="BH")
table(bh<0.05)
```

```
##
## FALSE  TRUE
##  3628    35
```

```
#output[which(bh<0.05),]
```

Question 2.4 Perform the q-value Fdr correction on your $p-$value vector. The qvalue function is provided in the qvalue package on Bioconductor. Make sure you have the latest version of R (>3.5) installed. Follow the instructions on the homepage for installation https://www.bioconductor.org/packages/release/bioc/html/qvalue.html and call the package using

```
library(qvalue)
```

Use $\alpha = 0.05$ as your significance threshold. How many methylation sites would be considered as Non-Null after $q-$value Fdr correction?

*Reply: Using the $q-$value correction we detect 39 methylation sites to be significant.

```
qobj = qvalue(p=pvec)
qvalues = qobj$qvalues
table(qvalues<0.05)
```

```
##
## FALSE  TRUE
##  3624    39
```

```
#output[which(qvalues<0.05),]
```

Further we see that the estimated proportion of Null variables (pi0) is 0.753487, which is equivalent to (1-0.753487) = 0.246513 Non-Null variables. Given that the dataset contains 3663 variables this would be around 900 potential signal variables. This indicates that there is a strong signal and many methylation sites are potentially associated with ageing, but given our sample size of n=409 we can only detect 39 methylation sites.

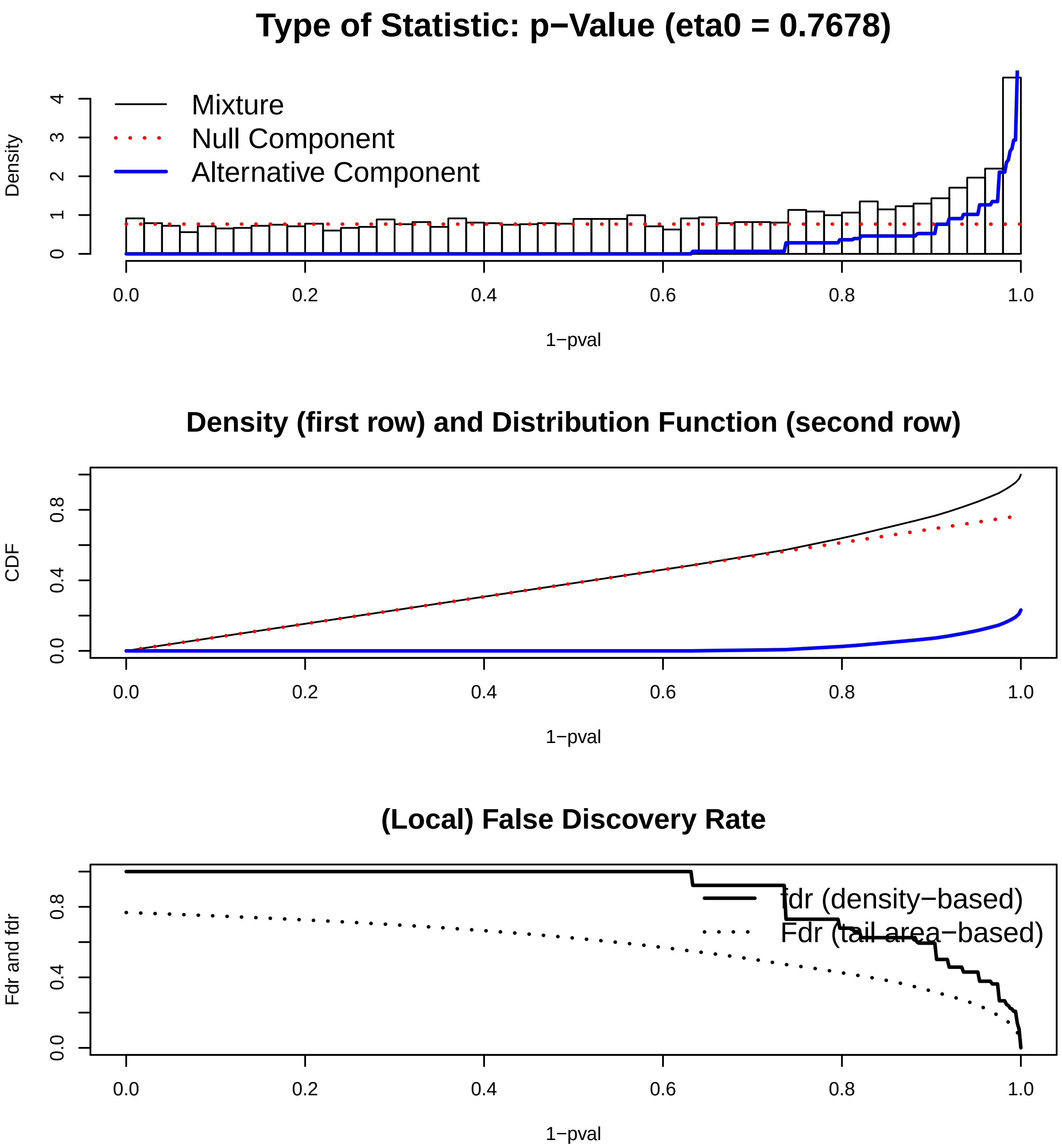
```
qobj$pi0
```

```
## [1] 0.753487
```

Question 2.5 Perform the local fdr correction on your $p-$value vector using the fdrtool R package. Use the option statistic='pvalue' to model the $p-$value vector and extract the local fdr estimate in the $\$lfdr$ value. How many methylation sites would be considered as Non-Null after local fdr correction at a level of $\alpha = 0.2$?

Reply: Using the local fdr correction we declare 164 methylation sites as discoveries.

```
library(fdrtool)
lfdr_out = fdrtool(x=pvec, statistic="pvalue", verbose=FALSE)
```



Type of Statistic: p−Value (eta0 = 0.7678)

Density (first row) and Distribution Function (second row)

(Local) False Discovery Rate

```
table(lfdr_out$lfdr<0.2)
```

```
##
## FALSE  TRUE
##  3499   164
```

```
#output[which(lfdr_out$lfdr<0.05),]
```

Question 2.6 Look at the top graph of the mixture models that the fdrtool package has as ouput. Where is the Null-distribution and where is the Non-Null distribution? Would you consider this to be a good model fit?

Reply: In red we can see the Null distribution which is a uniform distribution and in blue we see the Non-Null distribution. The underlying histogram shows the actual observed data. We indeed see a good fit of the observed data with our fitted mixture distribution.

Question 2.7 The fdrtool estimates the proportion of Null variables. Access this information in the $param value. What is the proportion of Null variables?

Reply: The estimated proportion of Null variables (pi0) is 0.7677763, which is equivalent to (1-0.753487) = 0.2322237 Non-Null variables.

```
lfdr_out$param
```

```
##         cutoff N.cens      eta0    eta0.SE
## [1,] 0.3688585   1775 0.7677763 0.01308332
```

Question 2.8 Compare the 4 ways of how to perform multiple testing correction and discuss which one is the most conservative.

Reply: The most conservative approach is the Bonferroni correction. Regarding the FDR approaches, the Benjamini-Hochberg approach is more conservative than the q−value, since it assumes that a priori there is no signal ($pi0 = 1$)*. In contrast, the q−value estimates pi0 from the data, allowing* $pi0 < 1$*; in this particular analysis the propostion of Null variables is 0.753487. These tail-area Fdr estimates are essentially adjusted p−values, while the local fdr is the posterior that a variable is Null given its observed p−value.*

Part 3 (optional): Random p-values

In order to evaluate the Null-distribution of p-values of a linear regression model we perform a simulation study

Question 3.1 First simulate both x and y from the normal-distribution with mean 0 and variance 1, for each variable draw a 1000 random draws using the rnorm() function in R using

```
set.seed(1234)
x=rnorm(1000)
y=rnorm(1000)
```

Compute the linear regression of x on y and look at the p-value of the regression coefficient.

Reply: We fit the regression using the lm() function and look at the value $coefficients.

```
summary(lm(y~x))$coefficients
```

```
##               Estimate Std. Error  t value   Pr(>|t|)
## (Intercept) 0.01599006 0.03100479 0.5157287 0.60615816
## x           0.05571331 0.03109205 1.7918831 0.07345456
```

Question 3.2 Repeat this operation 10,000 times using a for loop. Save your $p-$values of each iteration in an object pvec defined as

```
pvec = rep(NaN, 10000)
```

Reply: We do this with the following for loop

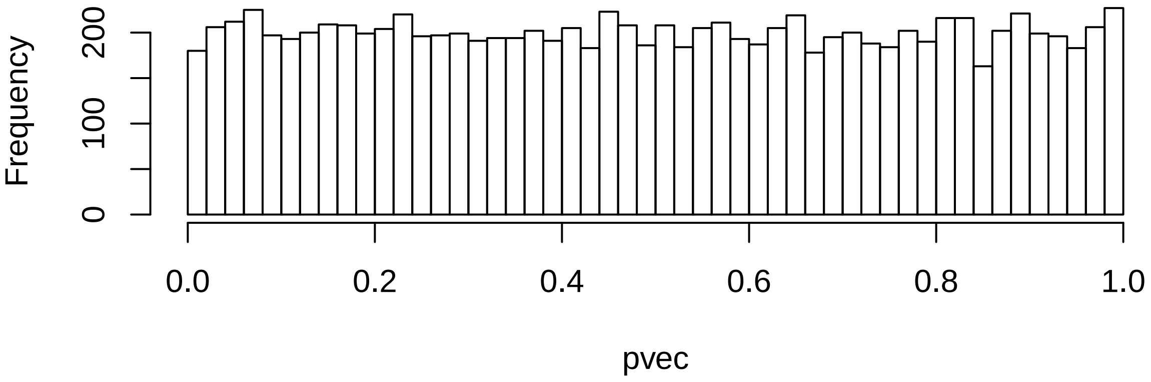
```
for(i in 1:10000){

    x=rnorm(1000)
    y=rnorm(1000)
    pvec[i]=summary(lm(y~x))$coefficients[2,4]

}
```

Question 3.3 Plot a histogram and discuss which distribution this vector of $p-$values has.

```r
hist(pvec, breaks=50, main="")
```



Reply: The Null-distribution of $p-$values follows a uniform distribution between 0 and 1.

Part 4 (Optional): FDR correction on GWAS summary data on Schizophrenia

Go to the webpage of the Psychiatric Genetics Consortium (PGC) https://www.med.unc.edu/pgc/results-and-downloads and download in the section SCZ2 the full SNP results by clicking on the link 'Download full SNP results'. After downloading and un-zipping the file we can load it into R using and check the dimension of the dataset

```r
data=read.csv("ckqny.scz2snpres", sep="\t")
dim(data)
```

```
## [1] 9444230      10
```

The file includes genome-wide data on over 9m SNPs. Looking at the first few lines of the data we find the column providing the p-values for the SNPS and save it as a vector pvec.

```r
head(data)
```

```
##   hg19chrc       snpid a1 a2     bp  info      or     se      p ngt
## 1     chr1   rs4951859  C  G 729679 0.631 0.97853 0.0173 0.2083   0
## 2     chr1 rs142557973  T  C 731718 0.665 1.01949 0.0198 0.3298   0
## 3     chr1 rs141242758  T  C 734349 0.666 1.02071 0.0200 0.3055   0
## 4     chr1  rs79010578  A  T 736289 0.649 0.98748 0.0193 0.5132   0
## 5     chr1 rs143225517  T  C 751756 0.853 0.99681 0.0164 0.8431   0
## 6     chr1   rs3094315  A  G 752566 0.881 0.99601 0.0149 0.7870  36
```

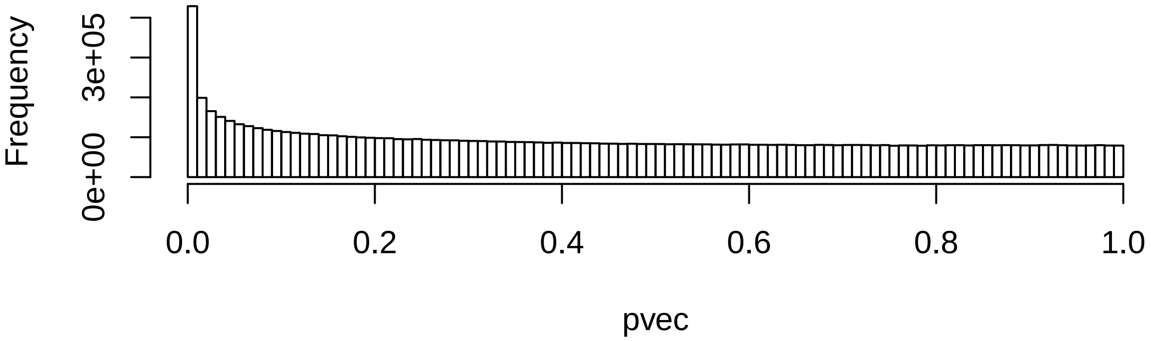
```r
pvec = data$p
N=length(pvec)
N
```

```
## [1] 9444230
```

Question 4.1 Plot a histogram of the p-value vector pvec. Discuss if this distribution can be modelled using a mixture distribution.

Reply: This is a perfect mixture distribution that consists of the following two distributions: - F_0 a uniform distribution of the Null or irrelavant SNPs and - F_A the sharp peak at around 0 that captures the Non-Null SNPs or the signal.

```r
hist(pvec, breaks=100, main="")
```



Question 4.2 Perform Bonferroni correction of the p-value vector pvec using the function p.adjust. How many SNPs are significant at a Bonferroni adjusted level of 0.05?

Reply: Following the Bonferroni correction there are 9,323 SNPs discovered at a level of 0.05.

```
p_bonferroni=p.adjust(p=pvec, method="bonferroni")
table(p_bonferroni<0.05)
```

```
##
##   FALSE    TRUE
## 9434907    9323
```

Question 4.3 Perform the Benjamini-Hochberg FDR correction of the p-value vector pvec using the function p.adjust. How many SNPs are significant at a Benjamini Hochberg FDR adjusted level of 0.05?

Reply: Following the Benjamini-Hochberg correction there are 12,4955 SNPs discovered at a level of 0.05.

```
p_bh=p.adjust(p=pvec, method="BH")
table(p_bh<0.05)
```

```
##
##   FALSE    TRUE
## 9319275  124955
```

Question 4.3 Perform the q-value FDR correction of the p-value vector pvec using the function qvalue in the qvalue package on bioconductor. Follow the instructions on the homepage for installation https://www.bioconductor.org/packages/release/bioc/html/qvalue.html and call the package using

```
library(qvalue)
```

How many SNPs are significant at a $q-$value level of 0.05? What is the proportion of Null SNPs?

Reply: Following the $q-$value correction there are 141,181 SNPs discovered at a level of 0.05.

```
qobj = qvalue(p=pvec)
qvalues = qobj$qvalues
table(qvalues<0.05)
```

```
##
##   FALSE    TRUE
## 9303049  141181
```

Looking at the pi0 object we see that the actual proportion of Null-SNPs is 0.8394974, which suggests that around 16% percent of the SNPs are associated with schizophrenia. This suggests that schizophrenia is a highly polygenic disease.

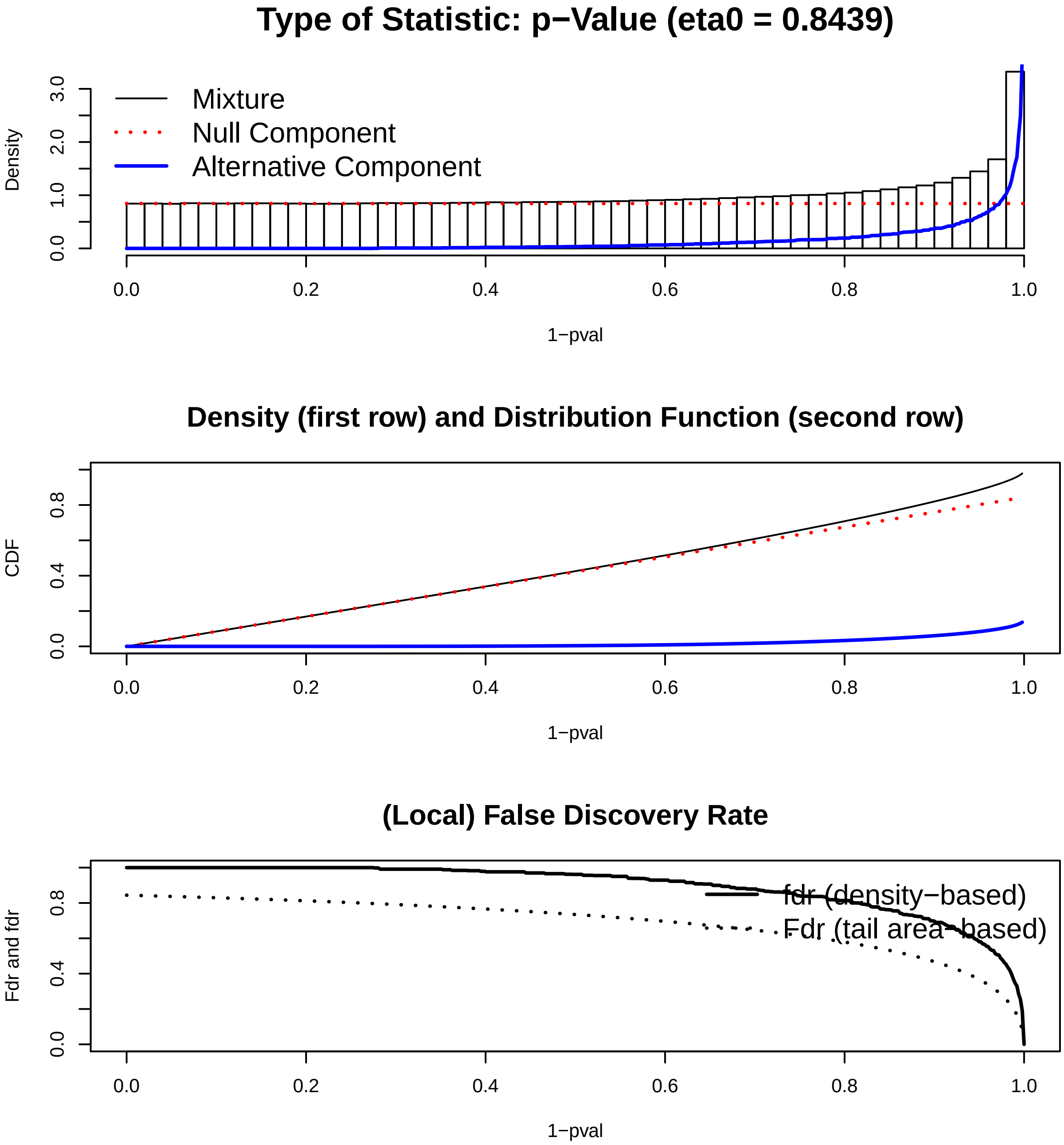
```
pi0 = qobj$pi0
pi0
```

```
## [1] 0.8394974
```

Question 4.4 Perform the local fdr correction on your $p-$value vector using the fdrtool R package. Use the option statistic='pvalue' and extract the local fdr estimate in the $\$lfdr$ value. How many SNPs would be considered as Non-Null after local fdr correction at a level of $\alpha = 0.2$?

Reply: At a local fdr level of 0.2 *there are 212,322 SNPs declared as discoveries.*

```
library(fdrtool)
lfdr_out = fdrtool(x=pvec, statistic="pvalue", verbose=FALSE)
```



```
table(lfdr_out$lfdr<0.2)
```

```
##
##   FALSE    TRUE
## 9231908  212322
```

Question 4.5 Look at the graph of the mixture models that the fdrtool package has as ouput. Would you consider this to be a good model fit? Further, what is the proportion of Null variables?

Reply: The observed data in the histogram fits nicely the fitted mixture model in black consisting of the red Null and blue Non-Null component.

```
lfdr_out$param
```

```
##         cutoff  N.cens      eta0      eta0.SE
## [1,] 0.7401648 2070932 0.8439201 0.0005181626
```

If you want to read more on the implications of FDR analysis in genome-wide association studies, take a look at the following recent article 'Association analyses based on false discovery rate implicate new loci for coronary artery disease' in Nature Genetics https://www.nature.com/articles/ng.3913 If you are accessing this article from outside of the College: - Click the link - Select Shibboleth Under 'Additional access options' - Search for and select Imperial College London - Login with your College username and password