Homework 2 - BIOS 7649

Dominic Adducci

2024-02-14

Question 1

Part A

Read in 8 CEL files in the directory celfiles.

Extracting data using the exprs() function.

```
##   High1 High2 High3 High4 Low1 Low2 Low3 Low4
## 1   197    81   106    69  174  103  100  100
## 2 13571  9974  9267  6591 7231 4831 4724 4724
## 3   248   137   100    94  194  147  137  137
## 4 13810  9895  9550  6866 7407 4886 4919 4919
## 5   123    63    60    76   77   79   71   71
## 6   173    73    61    68  116   81  103  103
```

Extracting data using the sampleNames() function:

```
## [1] "High1" "High2" "High3" "High4" "Low1"  "Low2"
```

Extracting data using the probeNames() function:

```
##
```

```
## [1] "1007_s_at" "1007_s_at" "1007_s_at" "1007_s_at" "1007_s_at" "1007_s_at"
```

Extracting data using mm() function:

```
##         High1 High2 High3 High4 Low1 Low2 Low3 Low4
## 370871    338   123   115   113  131  152  145  145
## 564482    831   435   393   397  695  791  852  852
## 1050513  1190   520   345   583  685  688 1033 1033
## 239977   1288   691   592   564  837  846  891  891
## 1141565  2740  1161  1035  1477 1540 1938 2269 2269
## 1131946   786   376   335   355  591  631  791  791
```

Extracting data using pm() function:

```
##         High1 High2 High3 High4 Low1 Low2 Low3 Low4
## 369707    368   220   216   248  405  379  476  476
## 563318   1350   687   590   585 1455 1626 1765 1765
## 1049349  1414   674   599   586 1464 1422 1701 1701
## 238813   3838  1830  1472  1446 2980 3205 3701 3701
## 1140401  3018  1338  1165  1445 2208 2616 3151 3151
## 1130782  1652   783   668   700 1249 1520 1784 1784
```

Extracting data using pData() function:

```
##                         FileName Target
## High1 High_1_HG-U133_Plus_2.CEL   High
## High2 High_2_HG-U133_Plus_2.CEL   High
## High3 High_3_HG-U133_Plus_2.CEL   High
## High4 High_4_HG-U133_Plus_2.CEL   High
## Low1   Low_1_HG-U133_Plus_2.CEL    Low
## Low2   Low_2_HG-U133_Plus_2.CEL    Low
```

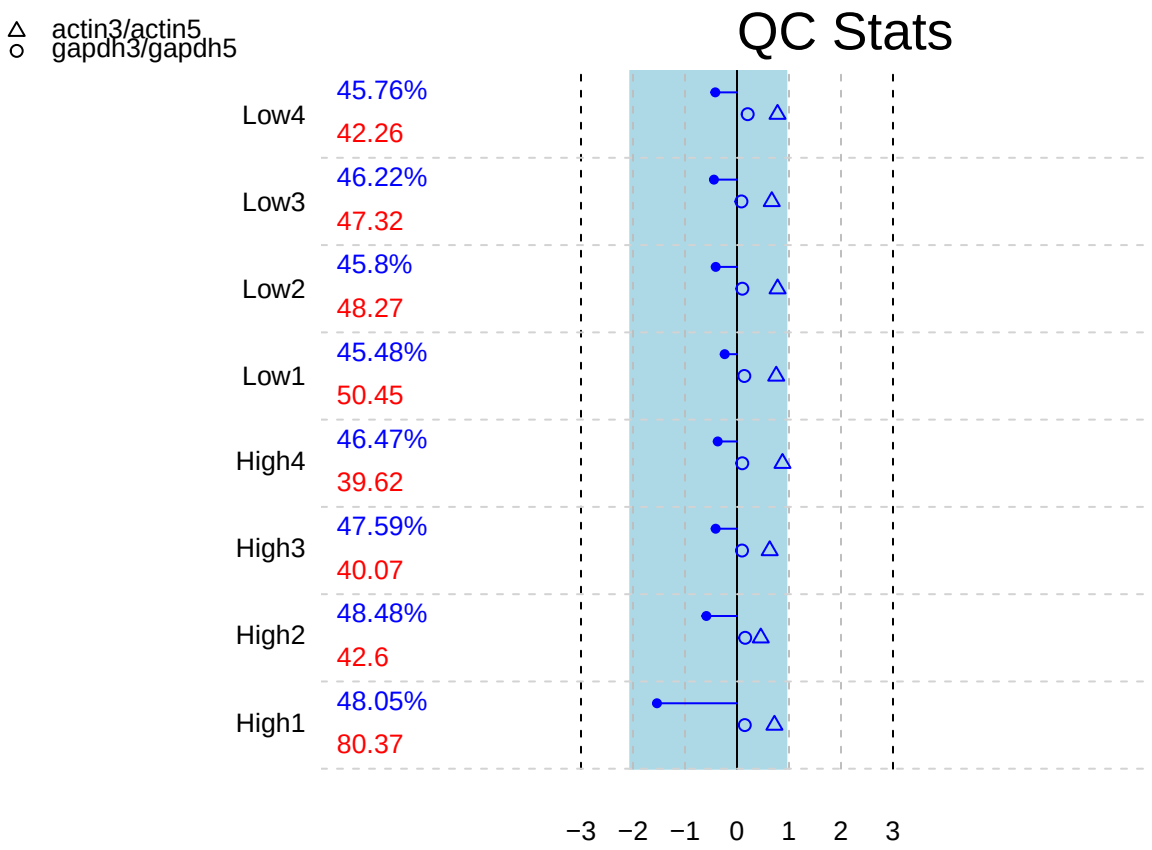
Part B

Plot the raw microarray images using **image()** on the object **Data**. Comment on what you see in these plots.

All plots look relatively similar with the exception of low 1, which has some light spots, and low 4, which appears to had something on the plate, and was not read correctly.

Part C

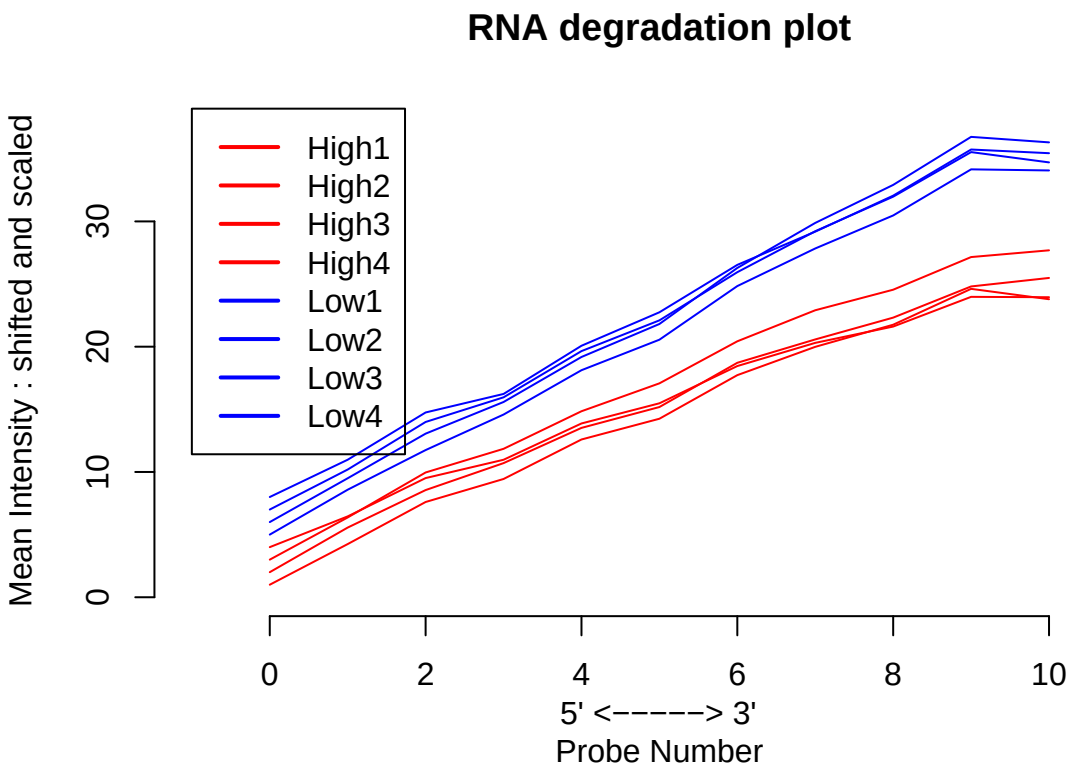
Plot quality control metrics using **qc.affy()** and **plot.qc.stats()** fom the **simpleaffy** package. Comment on what you see in the plot. A description of **simpleaffy** is in Wilson & Miller (2005).



All scale factors are within 3 folds of each other. Because the ends of all lines are within the blue region the scale-factors are compatible. Additionally, each line being colored blue indicates that these are OK.

Part D

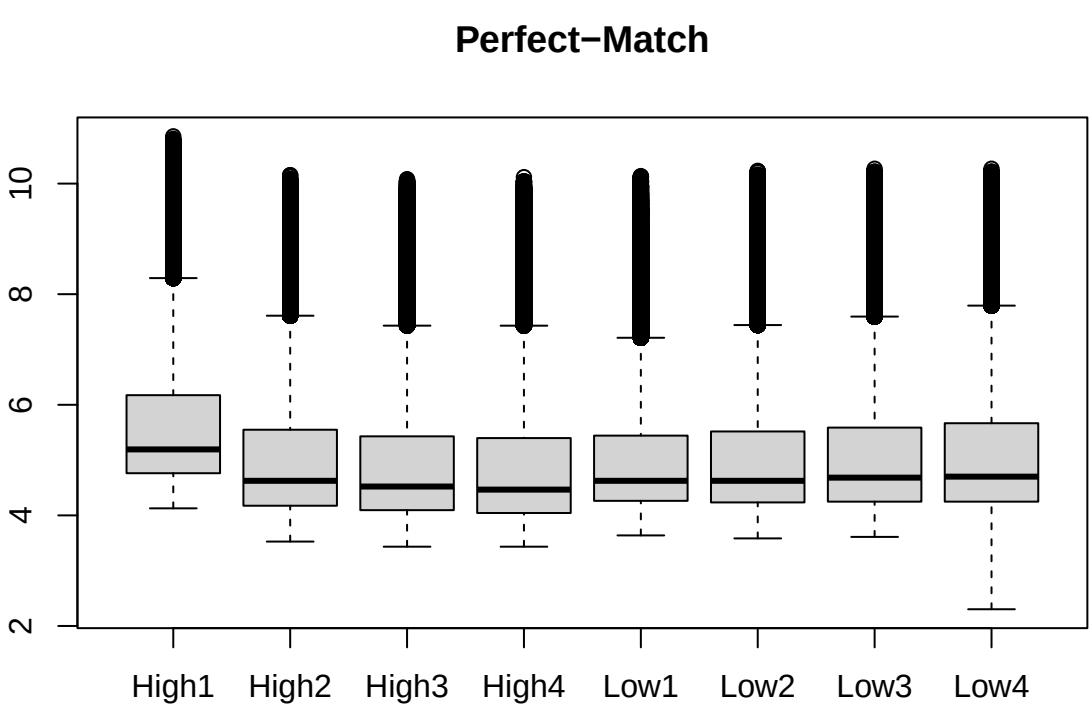
Plot he mean intensity from 3' to 5' end of the target mRNA using **AffyRNAdeg()** and **plotAffyRNAdeg()**. Comment on what you see in the plot.

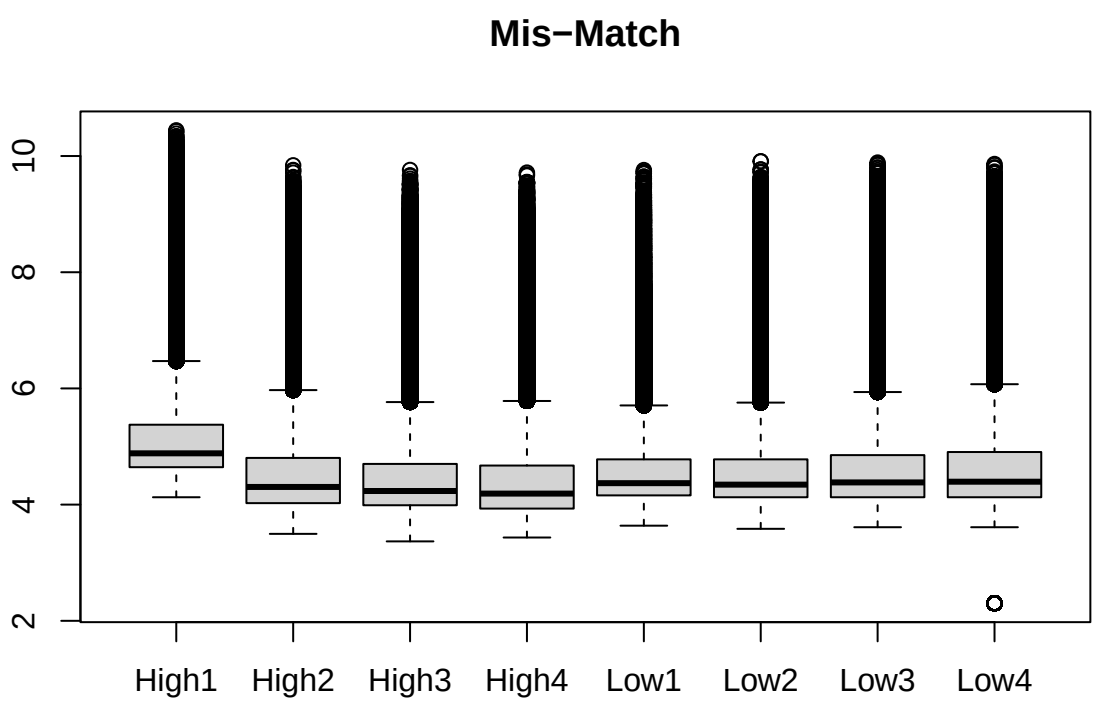


The mean intensity moving towards the 3' end is higher for the 'Low' plates compared to the 'High' plates. This pattern is consistant across all plates.

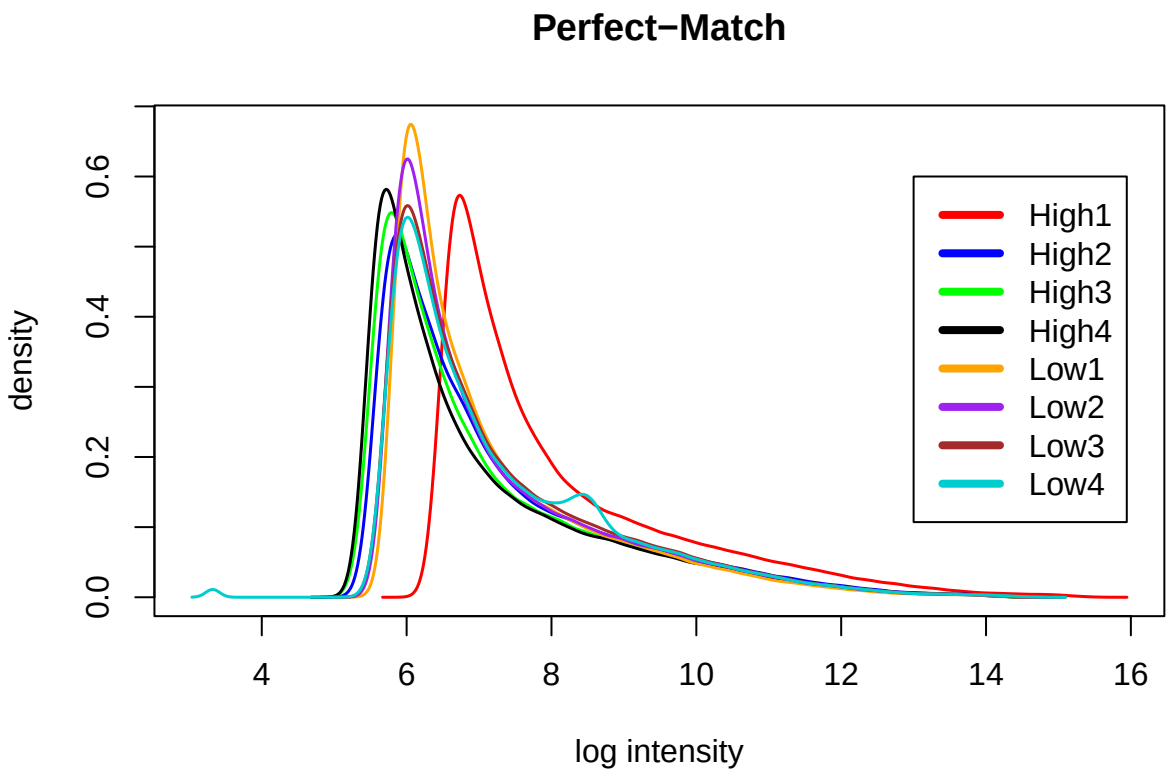
Part E

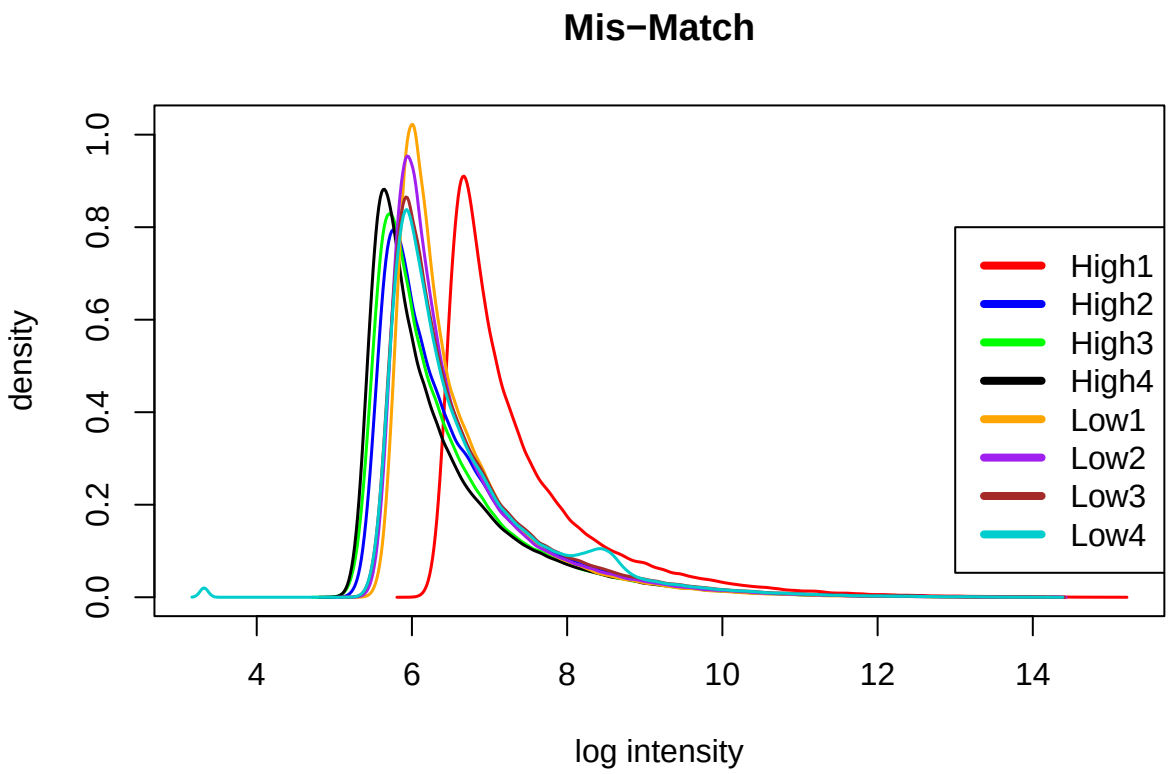
Use **boxplot()** and **plotDensity.AffyBatch()** to examine the distribution of intensity values for the perfect-match and mis-match probes separately. What patterns do you see? (Make sure you are plotting the **log** transformed data).


Perfect−Match
2
4
6
8
10
High1
High2
High3
High4
Low1
Low2
Low3
Low4



The means are similar between the perfect-match and mis-match probes, while the variance is greater for the perfect-match probes. For the mis-match probes, chip "Low4" has a noticeable outlier far below the mean.


Perfect−Match
density
0.0
0.2
0.4
0.6
4
6
8
10
12
14
16
log intensity
High1
High2
High3
High4
Low1
Low2
Low3
Low4



The mis-match probes have a higher peak in density compared to the perfect-match probes, although both have similar shapes. Chip "Low4" has noticeable secondary peaks on both the left and right hand side of the main peak.

Part F

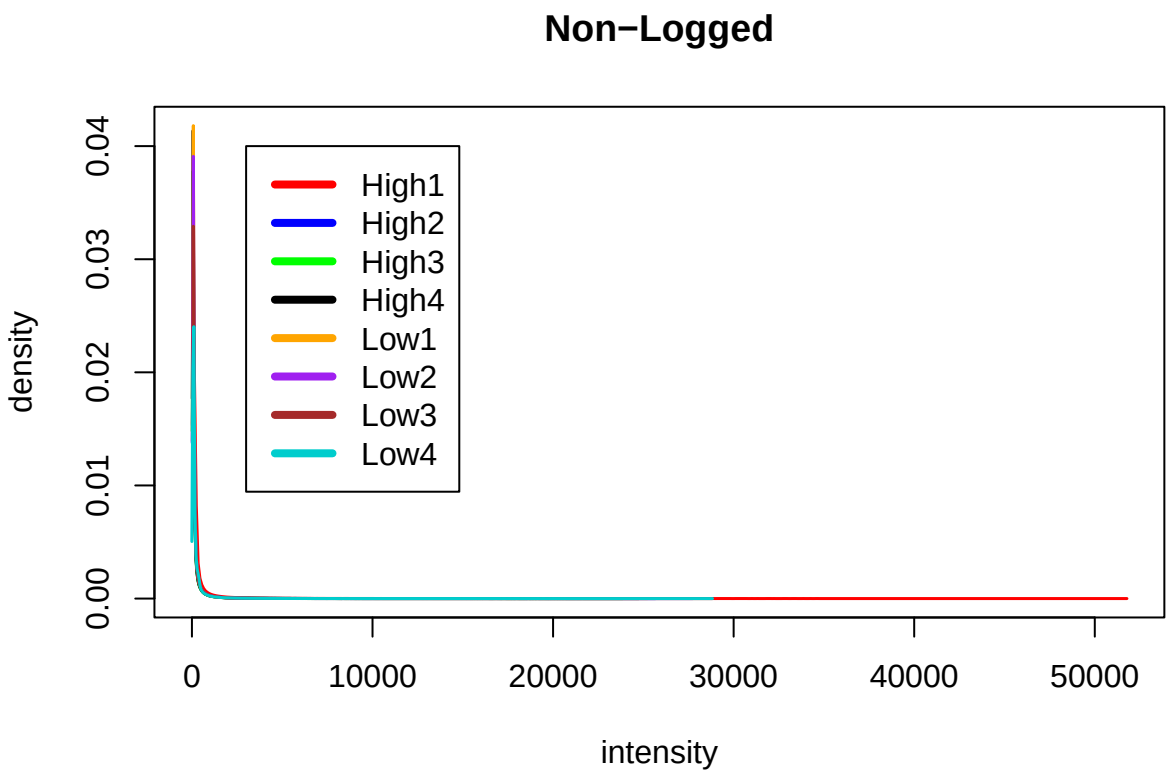
Based on the summaries and figures you generated, would you recommend that one or more chips be remove from the analysis?

Chip "Low4" should be removed, as there are noticeable abberations in the density plots.

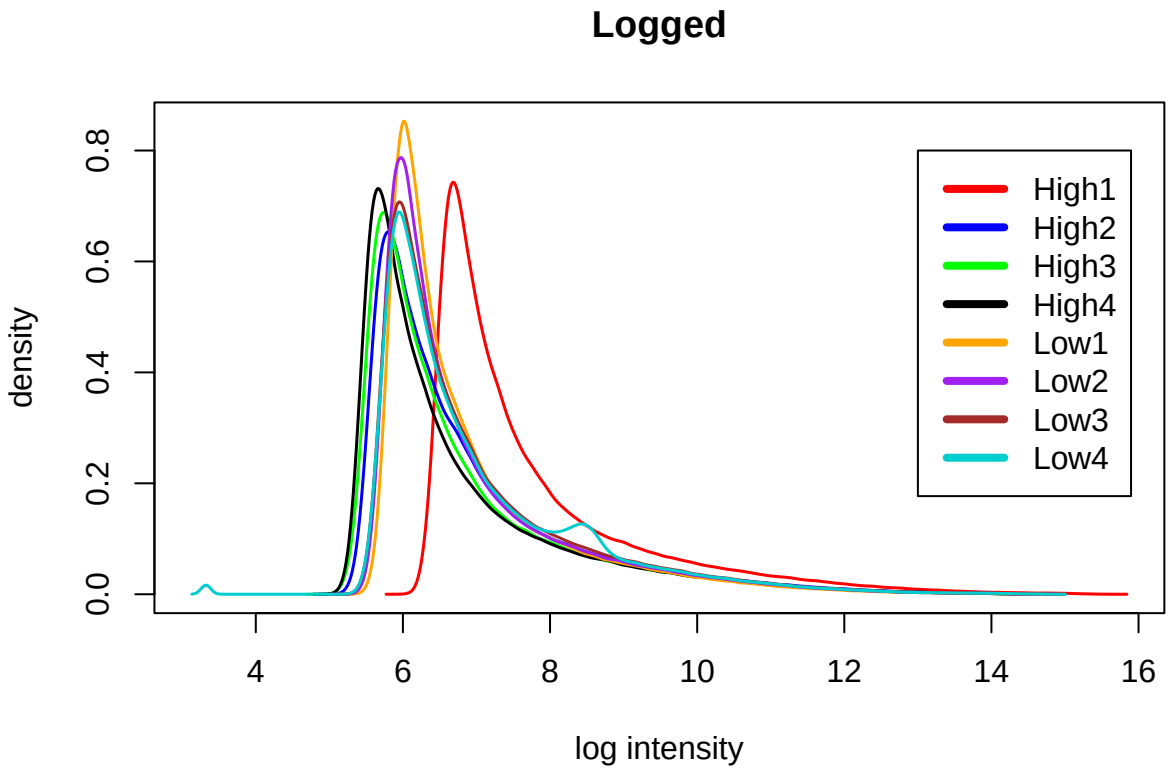
Question 2

Part A

Create log transformed data and plot the density before and after log transforming using **plotDensity.AffyBatch**. Comment on these plots.



Before log transforming the intensity values on the x-axis are much higher, and the lines have an exponential decay pattern, asymptotically approaching 0.



After log transforming the data the intensity scale is reduced, and the shapes are peaked. Additionally, this plot is far easier to read and interpret what is going on between chips.

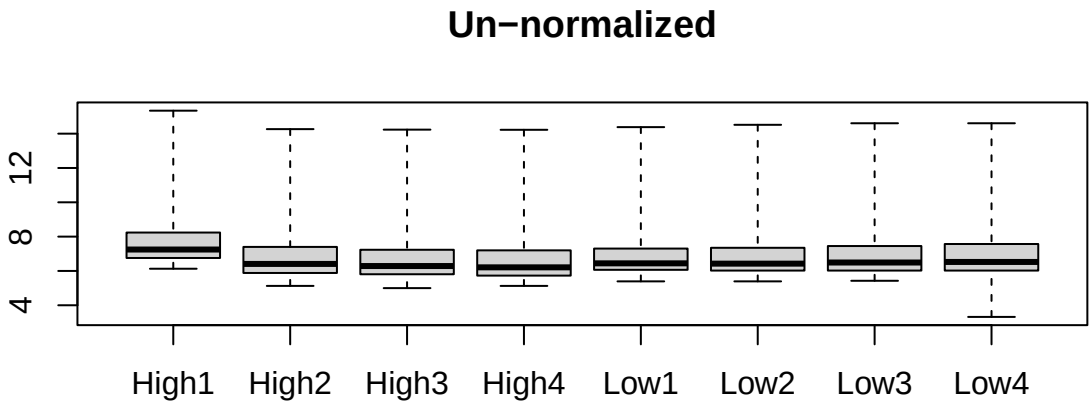
Part B

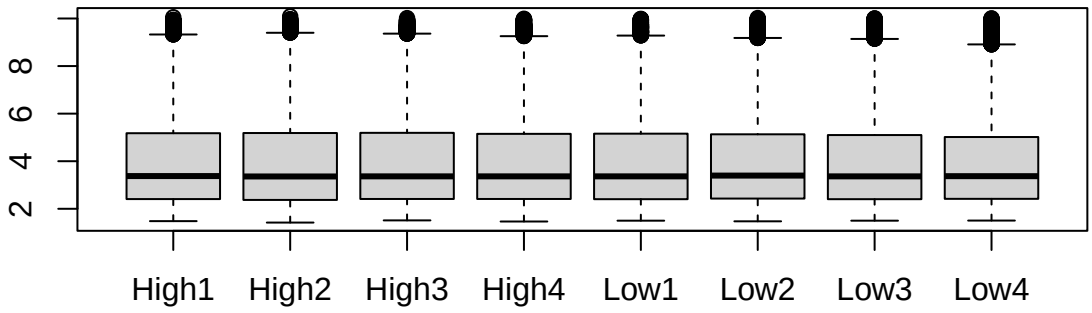
Plot MA plots using **MAplot()** and summarize your observations.

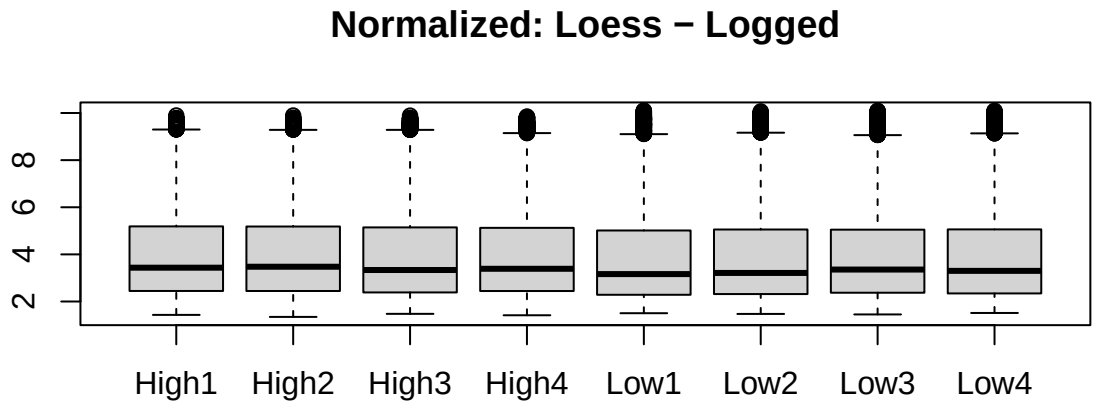
All chips have point clouds that are centered around 0 with the exception of chip "High1" and chp "Low4". Chip "High1" has the center slightly above 0, while chip "Low4" has two downward clouds in addition to the cloud centered around 0.

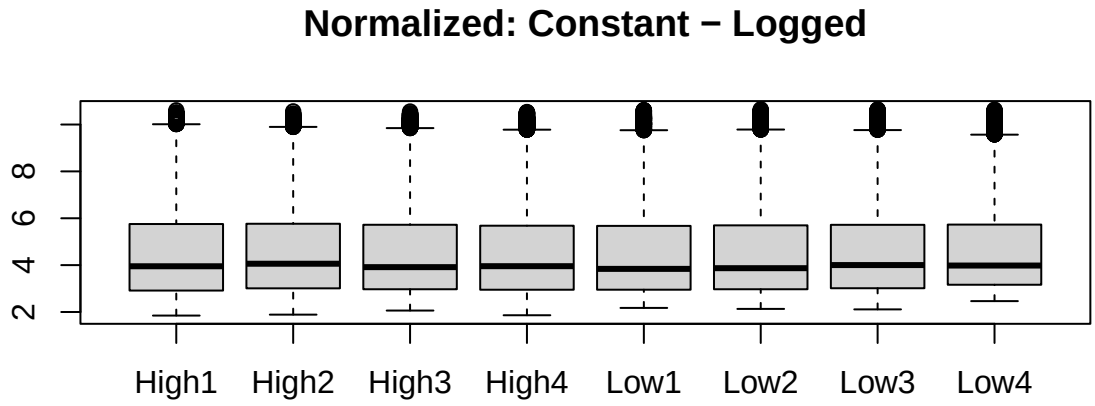
Part C

Use **expresso()** to try different **normalized.method** options, while keeping other arguments static. Make boxplots of un-normalized and normalized intensities. Try different **summary.method** options and **pm-correct.method** options.

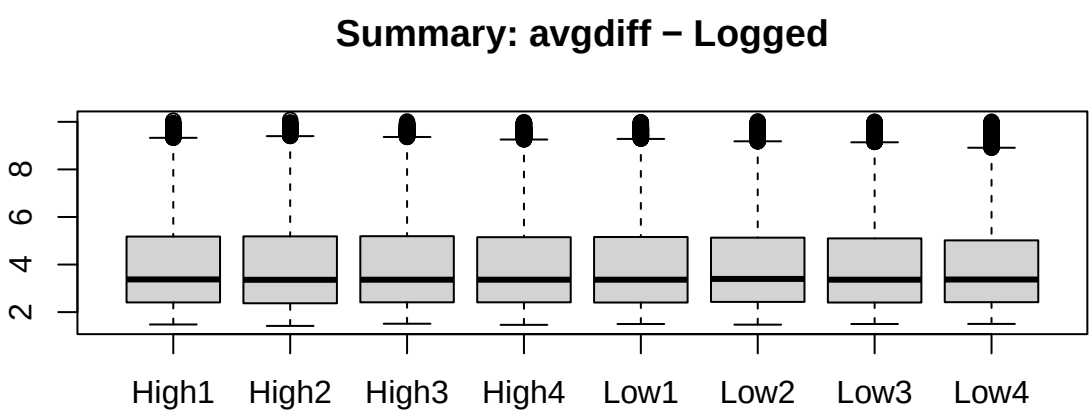

Un−normalized
4
8
12
High1
High2
High3
High4
Low1
Low2
Low3
Low4

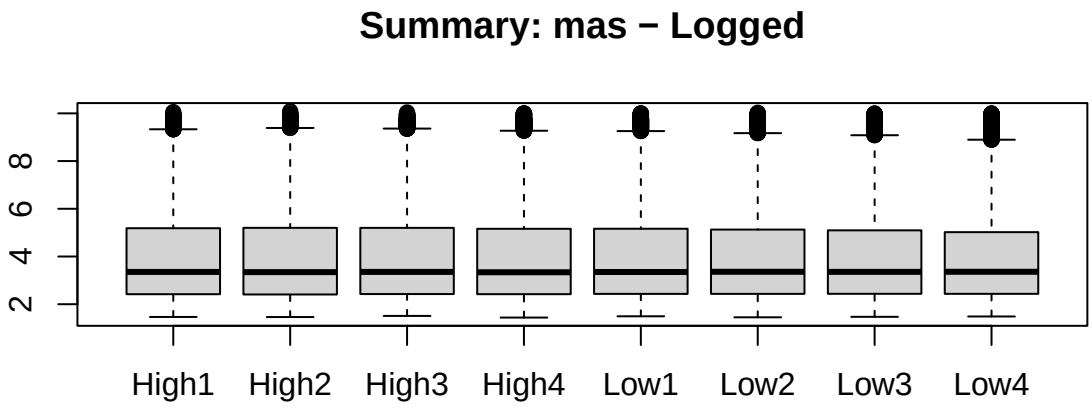

Normalized: Quantile − Logged
2
4
6
8
High1
High2
High3
High4
Low1
Low2
Low3
Low4

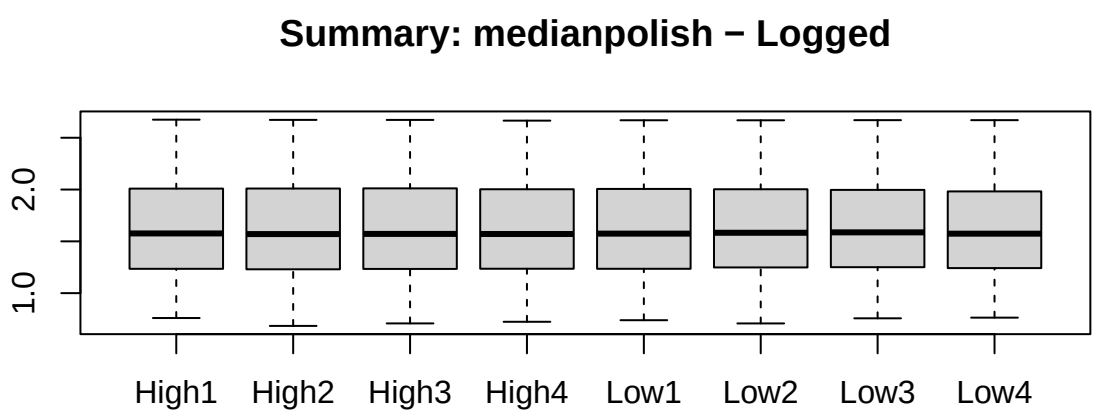




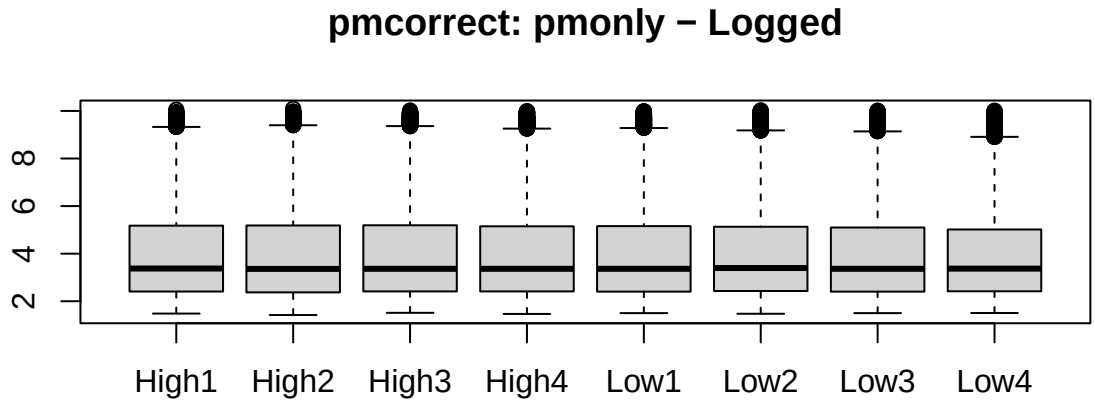
The un-normalized data set was plotted without being logged, all the normalized data sets were plotted after being logged. The means were the same between all of the normalized methods. The Loess method had the smallest variance, while the constant method tended to have the highest variance. All pmcorrect.methods were set to "pmonly", all summary.methods were set to "avgdiff", and bgcorrect.methods were set to "rma".

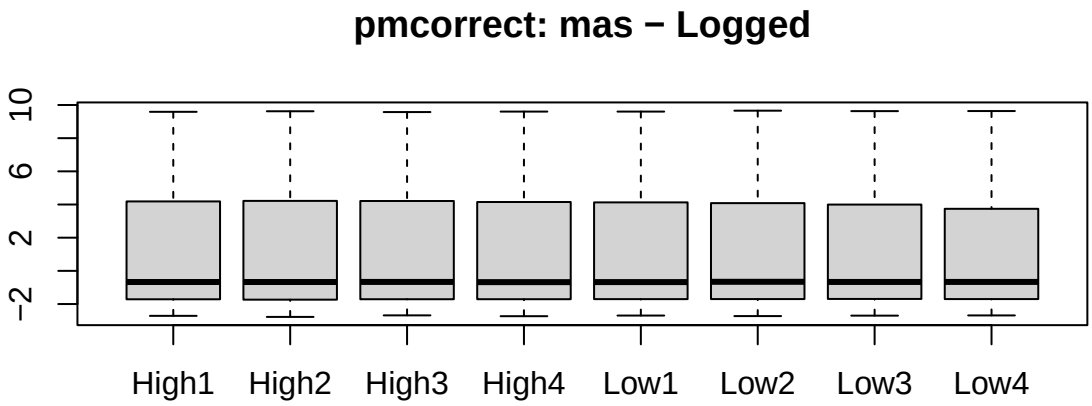

Summary: avgdiff − Logged
2
4
6
8
High1
High2
High3
High4
Low1
Low2
Low3
Low4

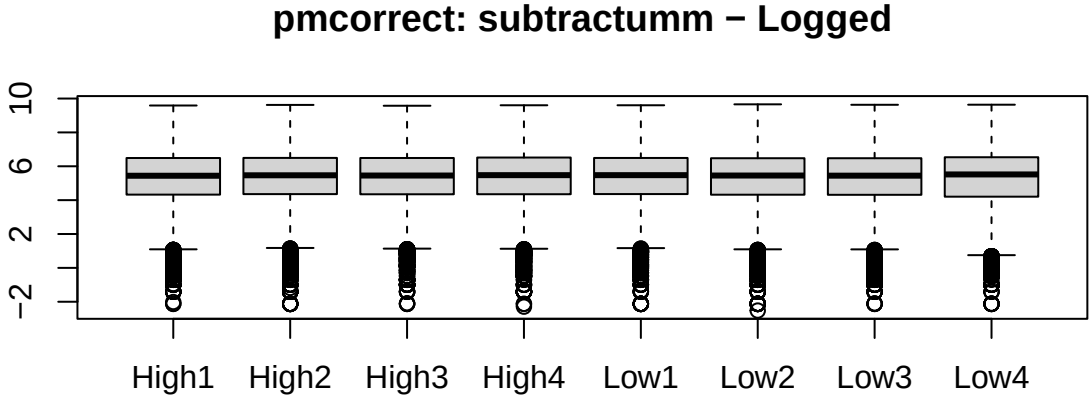

Summary: mas − Logged
2
4
6
8
High1
High2
High3
High4
Low1
Low2
Low3
Low4



Each summary method was logged before plotting. The avgdiff method and the mas methods performed very similarly, with the avgdiff method having slightly higher variance on a few chips. The medianpolish method had much less variance and the means were consistent across chips. The variances were also very similar across chpis. All pmcorrect.methods were set to "pmonly", all normalize.methods were set to "quantiles", and bgcorrect.methods were set to "rma".







All different pmcorrect.methods were logged before plotting. The pmonly method had the lowest variance, although there are more points that extend beyond the boxplot whiskers. The mas method had higher variance, but there were not points extending beyond the boxplot whiskers. The subtractumm method had a higher average than either of the other two methods, and had points extending noticeably below the mean. All summary.methods were set to "avgdiff", all normalize.methods were set to "quantiles", and bgcorrect.methods were set to "rma".

Part D

Get presentand absent calls Mas 5.0 using **mas5calls()**. How many probesets have at least one present call in each of the two groups? Use this as your filter for part e).

```
## Getting probe level data...
## Computing p-values
## Making P/M/A Calls
```

```
## [1] 54675
```

54,675 probesets had at least one P.

CODE

```
library(tidyverse)
library(kableExtra)
library(affy)
library(gcrma)
library(genefilter)
library(simpleaffy)


### START QUESTION 1 CODE ###

## START QUESTION 1 PART A CODE ##

# Reading the 8 CEL files in
# Desktop Location:
# Laptop Location: ""C:/Biostatistics Masters Program/Spring 2024/SMG-BIOS7659/Homework 2/targets.txt""

pd <- read.AnnotatedDataFrame("C:/Biostatistics Masters Program/Spring 2024/SMG-BIOS7659/Homework 2/targ
                              as.is=TRUE)

# Reading in CEL files

Data <- ReadAffy(filenames=pData(pd)$FileName,
                 phenoData=pd,sampleNames=sampleNames(pd))


head(exprs(Data))


head(sampleNames(Data))


head(probeNames(Data))


head(mm(Data))


head(pm(Data))


head(pData(Data))

## FINISH QUESTION 1 PART A CODE ##


image(Data)


## START QUESTION 1 PART C CODE ##

quality_data <- qc.affy(Data)
```

```
plot.qc.stats(quality_data)

## FINISH QUESTION 1 PART C CODE ##


## START QUESTION 1 PART D CODE ##

rna_deg <- AffyRNAdeg(Data)

plotAffyRNAdeg(rna_deg, cols = c(rep("red",4), rep("blue",4)))
legend(-1, 39, sampleNames(Data),
       lty = 1, col = c(rep("red",4), rep("blue",4)),lwd = 2)


boxplot(log(pm(Data)))
title("Perfect-Match")

boxplot(log(mm(Data)))
title("Mis-Match")


plotDensity.AffyBatch(Data,which = "pm",
                      col = c("red","blue","green","black",
                              "orange","purple","brown","cyan3"),
                      lty = 1, lwd = 1.5)
title("Perfect-Match")
legend(13,0.6,sampleNames(Data),col = c("red","blue","green","black",
                                        "orange","purple","brown","cyan3"),
       lty = 1, lwd = 4)

plotDensity.AffyBatch(Data,which = "mm",
                      col = c("red","blue","green","black",
                              "orange","purple","brown","cyan3"),
                      lty = 1, lwd = 1.5)
title("Mis-Match")
legend(13,0.8,sampleNames(Data),col = c("red","blue","green","black",
                                        "orange","purple","brown","cyan3"),
       lty = 1, lwd = 4)


### START QUESTION 2 CODE ###

## START QUESTION 2 PART A CODE ##

plotDensity.AffyBatch(Data, which = "both", log = FALSE,
                      col = c("red","blue","green","black",
                              "orange","purple","brown","cyan3"),
                      lty = 1, lwd = 1.5)
title("Non-Logged")
legend(3000,0.04, sampleNames(Data),col = c("red","blue","green","black",
                                            "orange","purple","brown","cyan3"),
       lty = 1, lwd = 4)
```

```
plotDensity.AffyBatch(Data, which = "both", log = TRUE,
                      col = c("red","blue","green","black",
                              "orange","purple","brown","cyan3"),
                      lty = 1, lwd = 1.5)
title("Logged")
legend(13,0.8,sampleNames(Data),col = c("red","blue","green","black",
                                        "orange","purple","brown","cyan3"),
       lty = 1, lwd = 4)


MAplot(Data, show.statistics = F)


## START QUESTION 2 PART C CODE ##

# Trying different normalization methods.
norm_quantile <- expresso(Data,bgcorrect.method = "rma",
                          pmcorrect.method = "pmonly",
                          summary.method = "avgdiff",
                          normalize.method = "quantiles",
                          verbose = FALSE)

norm_loess <- expresso(Data,bgcorrect.method = "rma",
                       pmcorrect.method = "pmonly",
                       summary.method = "avgdiff",
                       normalize.method = "loess",
                       verbose = FALSE)

norm_const <- expresso(Data,bgcorrect.method = "rma",
                       pmcorrect.method = "pmonly",
                       summary.method = "avgdiff",
                       normalize.method = "constant",
                       verbose = FALSE)



# Making boxplots of different normalizaton methods.
boxplot(Data)
title("Un-normalized")

boxplot(log(exprs(norm_quantile)))
title("Normalized: Quantile - Logged")

boxplot(log(exprs(norm_loess)))
title("Normalized: Loess - Logged")

boxplot(log(exprs(norm_const)))
title("Normalized: Constant - Logged")


# Trying different summary methods.
sum_avgdiff <- expresso(Data,bgcorrect.method = "rma",
                        pmcorrect.method = "pmonly",
```

```
                        summary.method = "avgdiff",
                        normalize.method = "quantiles",
                        verbose = FALSE)

sum_mas <- expresso(Data,bgcorrect.method = "rma",
                    pmcorrect.method = "pmonly",
                    summary.method = "mas",
                    normalize.method = "quantiles",
                    verbose = FALSE)

sum_median <- expresso(Data,bgcorrect.method = "rma",
                       pmcorrect.method = "pmonly",
                       summary.method = "medianpolish",
                       normalize.method = "quantiles",
                       verbose = FALSE)


# Making boxplots of different summary methods.
boxplot(log(exprs(sum_avgdiff)))
title("Summary: avgdiff - Logged")

boxplot(log(exprs(sum_mas)))
title("Summary: mas - Logged")

boxplot(log(exprs(sum_median)))
title("Summary: medianpolish - Logged")


# Trying different pmcorrect.method options

pmcorrect_pmonly <- expresso(Data,bgcorrect.method = "rma",
                             pmcorrect.method = "pmonly",
                             summary.method = "avgdiff",
                             normalize.method = "quantiles",
                             verbose = F)

pmcorrect_mas <- expresso(Data,bgcorrect.method = "rma",
                          pmcorrect.method = "mas",
                          summary.method = "avgdiff",
                          normalize.method = "quantiles",
                          verbose = F)

pmcorrect_sub <- expresso(Data,bgcorrect.method = "rma",
                          pmcorrect.method = "subtractmm",
                          summary.method = "avgdiff",
                          normalize.method = "quantiles",
                          verbose = F)



boxplot(log(exprs(pmcorrect_pmonly)))
title("pmcorrect: pmonly - Logged")
```

```r
boxplot(log(exprs(pmcorrect_mas)))
title("pmcorrect: mas - Logged")

boxplot(log(exprs(pmcorrect_sub)))
title("pmcorrect: subtractumm - Logged")

## FINISH QUESTION 2 PART C CODE ##


pres_abs <- mas5calls(Data)

present <- function(x){
  rowSums(x == "P") > 0
}

present_abs_results <- length(present(exprs(pres_abs)))

present_abs_results
```