Homework 7 - BIOS 7649

Dominic Adducci

All analysis were performed in R version 4.3.1

Question 1: DNA Methylation QC and Normalization (Illumina 450K)

Part A

In clinical manuscripts, the first table often includes summaries of clinical and demographic data (e.g., disease status, race, age, etc.). Create a table with this information (means, standard deviations, etc.), using **pData(rgSet)** to extract relevant information. How many unique subjects are there?

Table 1: Demographic Data

Age(mean,sd)	76.67(6.47)
Height(mean,sd)	166.3(14.25)
Weight(mean,sd)	61.9(7.23)
Patient Race	
White	2
African American	1
Sample Type (n = 6)	
With Cancer	3
Without Cancer	3
Patient Sex (n = 3)	
Male	1
Female	2

There are three unique subjects and each subject provides two a sample from a tumor and a samples from solid normal tissue.

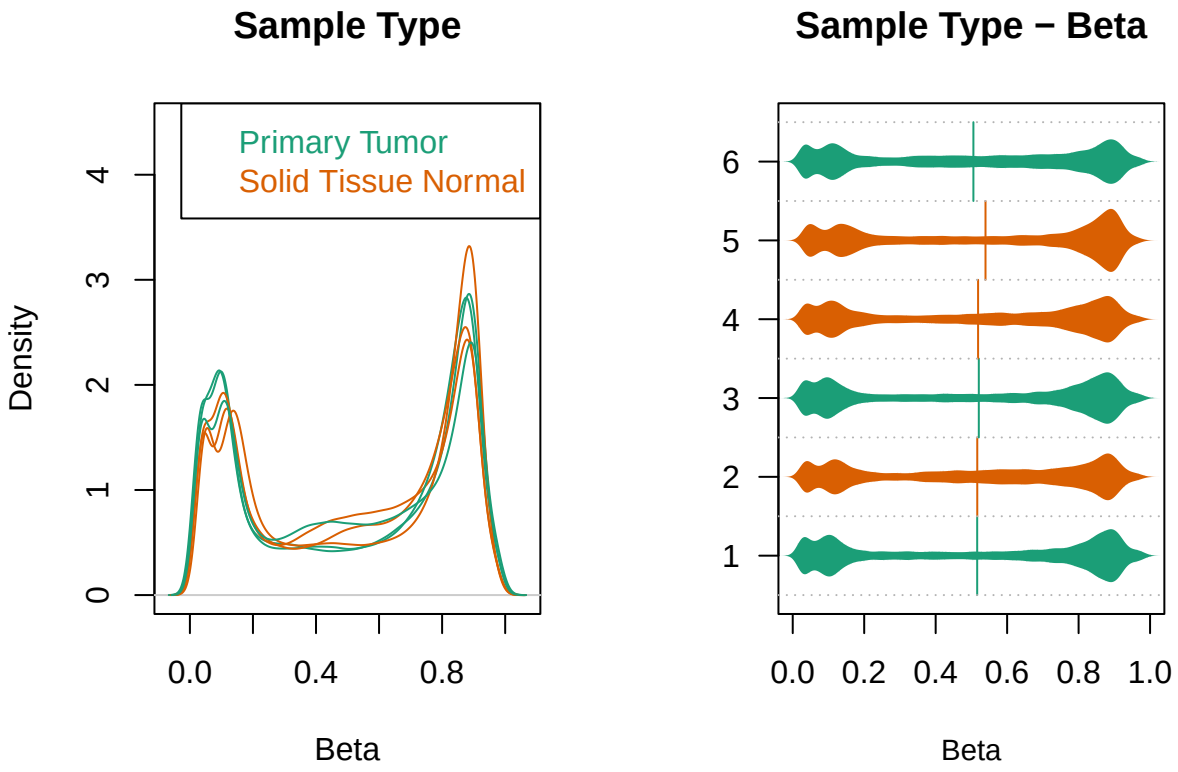
Part B

From the array annotation information given by **getManifest(rgSet)**, how many Type I and II probes are there? What is the difference between Type I vs II probes?

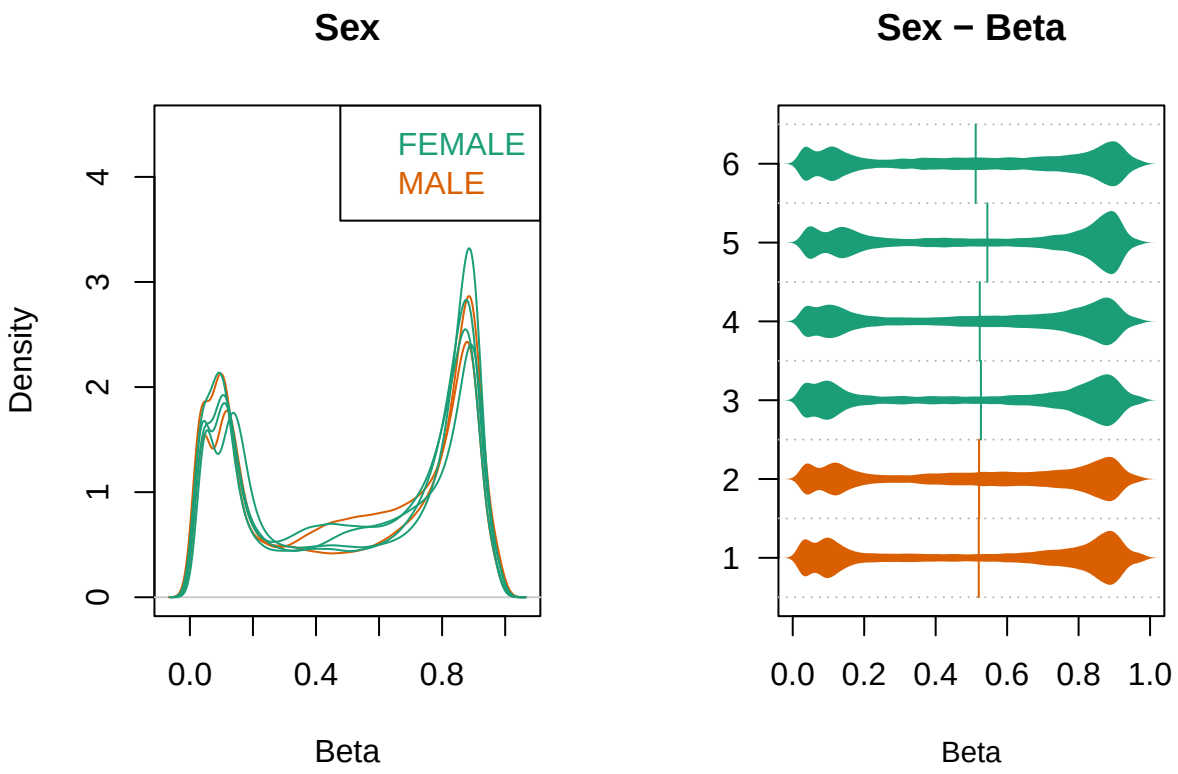
There are 135,476 Type I probes and 350,036 Type II probes. Type I probes have separate probes sequences for methylated and unmethylated CpG cites, while Type II probes use only one probe for each CpG site. The advantage of the Type II probes are that you can fit more probes on BeadChip, but there is lower dynamic range, bias, less reproducible, and difficulty comparing with results that utilized Type I probes.

Part C

Use **densityPlot()** and **densityBeanPlot()** to display QC plots. In the information from the "targets" file, use "id" for **sampNames** and repeat the QC plots on "sample_type" and "Sex" for **sampGroups** to see if there are differences in cancer versus normal subjects or by sex. Do you see any differences in the beta values between sample type and sex using the QC reports? Are there any samples that appear to be problematic?



The plots above show the density plot and density bean plot for samples by sample type (normal tissue or tumor). The density plot should ideally be bimodal, showing peaks for the methylated and unmethylated regions. This appears to be satisfied in the density plot. In the bean plot samples 5 and 6 are noticeably different. As these came from the same subjects this may be due to a difference within that subject.



The interpretation of these plots is that same as above. Interestingly, the subject who provided samples 5 and 6 was recorded as "FEMALE", which would be that same as the subject who provided samples 3 and 4. However, the bean plot suggests that samples 3 and 4 are closer to the subject who provided samples 1 and 2, who was reported as "MALE".

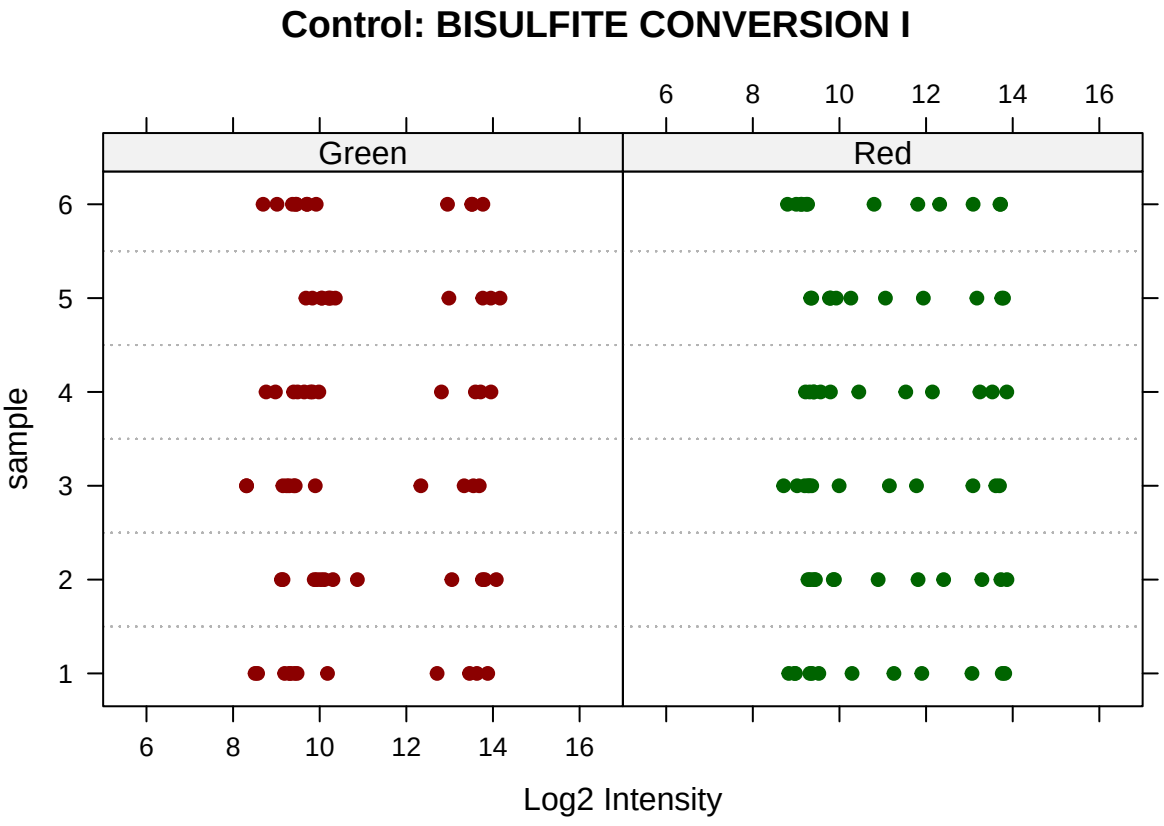
Part D

Describe their purpose of the different control probes on the array (see link above and **help(qcReport)**). Use **controlStripPlot()** to display the intensity values for the "BISULFITE CONVERSION I" and "NEGATIVE" probes. What do the ranges of these intensity values tell us about the quality of the data?

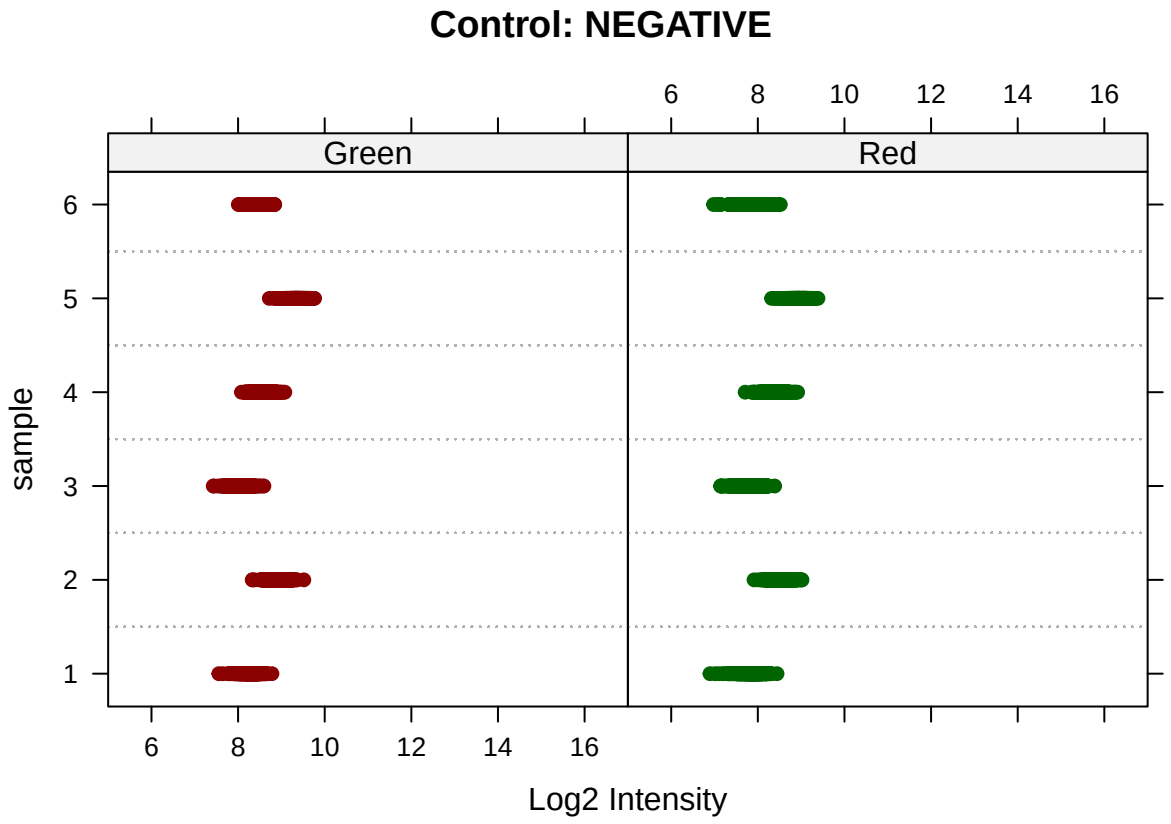
From the Illumina help guide there are thirteen different control probes. The summaries below are essentially verbatim from the help document.

- Staining Controls: These examine the efficiency of staining for both the red and green channels.These are independent of the hybridization and extension step.
- Extension Controls: These test the extension efficiency of A,T,C, and G nucleotides from a hairpin probe. These are sample independent, and should be monitored in the red (A,T) and green (C,G) channels.
- Hybridization Controls: These test the overall performance of the Infinium Assay using synthetic targets instead of amplified DNA. These should be monitored in the green channel only.
- Target Removal Controls: These test the efficiency of the stripping step after the extension reaction. These should result in low signal compared to the hybridization control, indicating that the targets were removed efficiently after extension. These controls should be monitored in the green channel only.
- Bisulfite Conversion Control: These assess the efficiency of bisulfite conversion of the genomic DNA.

- Bisulfite Conversion I: These are of the Infinium I probe design and use allele-specific single base extension to monitor efficiency of bisulfite conversion. These are monitored in both the red and green channels.
- Bisulfite Conversion II: These are of the Infinium II probe design and use a single base extension to monitor efficiency of bisulfite conversion. Depending on the status of conversion either the red or green channel will show elevated signal.
- G/T Mismatch Controls: These check for non-specific detection of methylation signal over unmethylated background. PM controls correspond to A/T perfect match and should show a high signal, MM controls correspond to G/T mismatch and should show low signal.
- Specificity Controls: These check for non-specific binding and should be monitored in both the green and red channel.
- Specificity I: These are designed to monitor potential non-specific primer extension for both Infinium I and Infinium II assay probes.
- Specificity II: These are designed to monitor allele-specific extension for Infinium I probes.
- Negative Controls: These target bisulfite-converted sequences that do not contain CpG dinucleotides. The mean signal of these probes define the system background. These should be monitored in both the green channel and red channel.
- Non-polymorphic Controls: These test the overall performance of the assay, form amplification to detection. These let you compare assay performance across different samples.



The above plot shows the quality control data for Bisulfite Conversion I controls. The range between samples are relativly similar, although 2 and 5 are slightly higher in the green channel, meaning there may have been higher bisulfite conversion.



The negative control probes target the sequences which do not have CpG sites, and provide the background noise. The ranges between samples are relatively similar, although sample 2 and 5 ares lightly higher in both channels, suggesting more noise in these samples.

Part E

Illumina also reports detection p-values, how are these calculated? Using the function **detectionP()**, which sample had the largest percentage of detection p-values ≥ 0.05? How many probes have average detection p-value ≥ 0.05 across the 6 samples?

Table 2: P-Values >= to 0.05

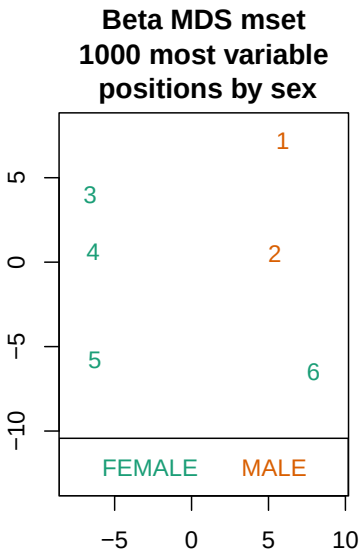
Sample	Percentage p-values >= 0.05
1	0.0002739
2	0.0011678
3	0.0002101
4	0.0010154
5	0.0014026
6	0.0011411

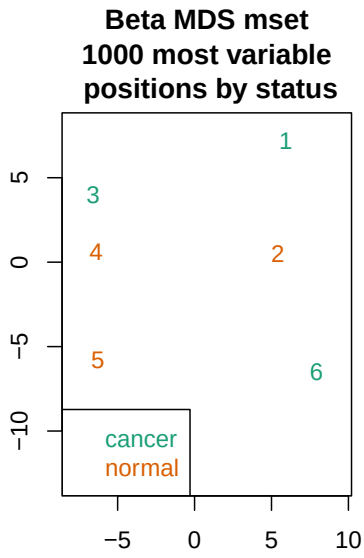
Sample 5 had the largest percentage of detection p-values ≥ 0.05.

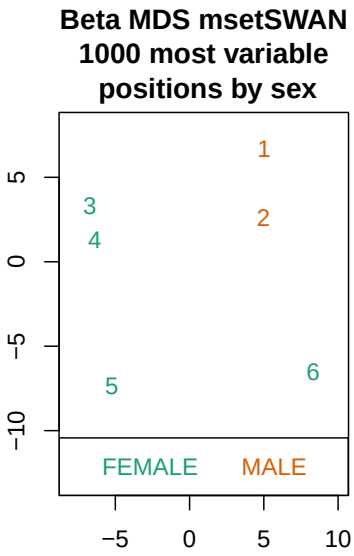
There are 853 probes that have an average detection p-value ≥ 0.05 across the 6 samples.

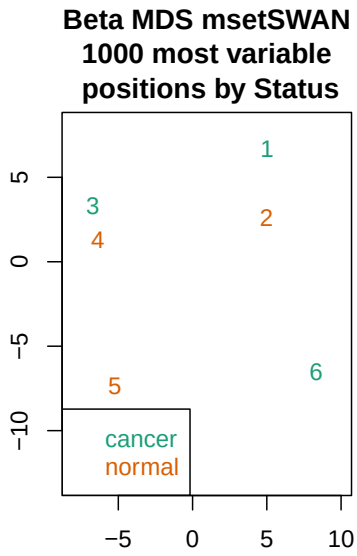
Part F

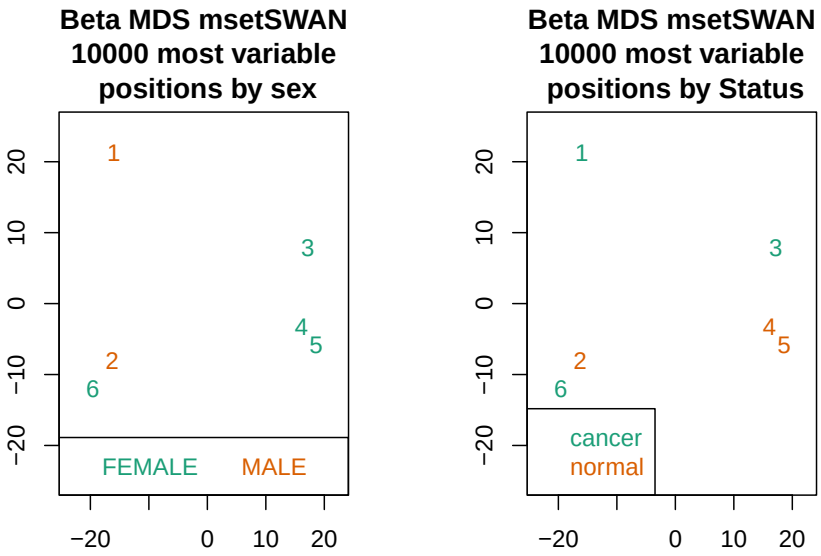
Use multidimensional scaling (MDS) plots to show how samples group by sex or cancer status with **mdsPlot()**. What do you conclude? Are conclusions different if you take more positions with the most methylation variability (1000, vs 10000 positions)? Or by using the raw data **mset** compared to the SWAN normalized data **msetSWAN**.







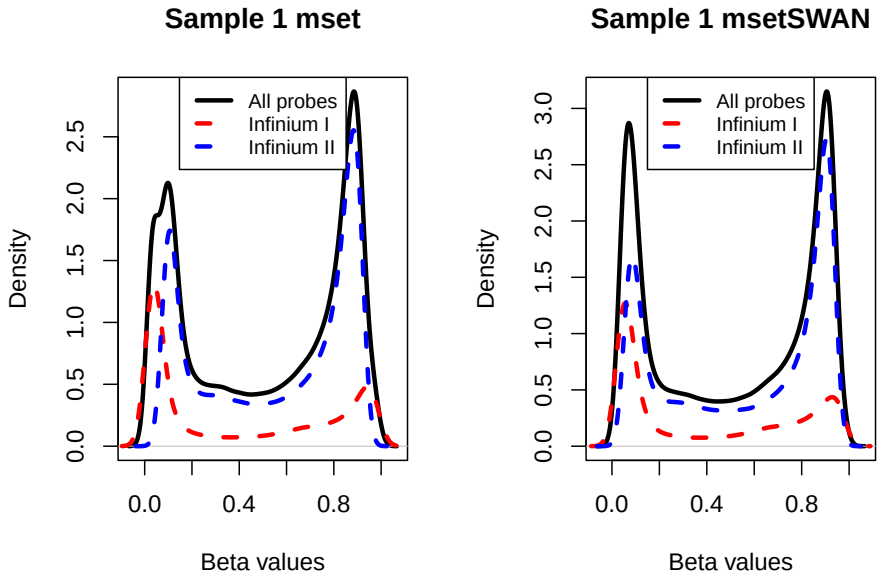


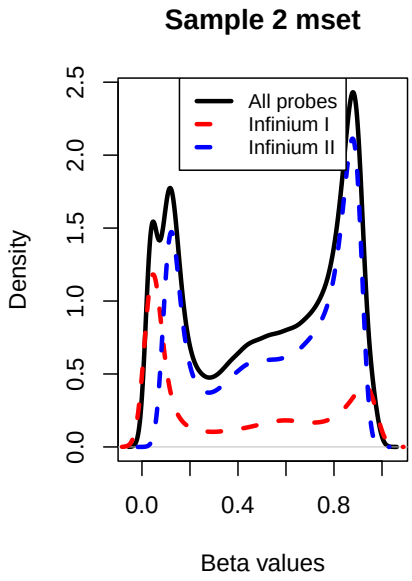


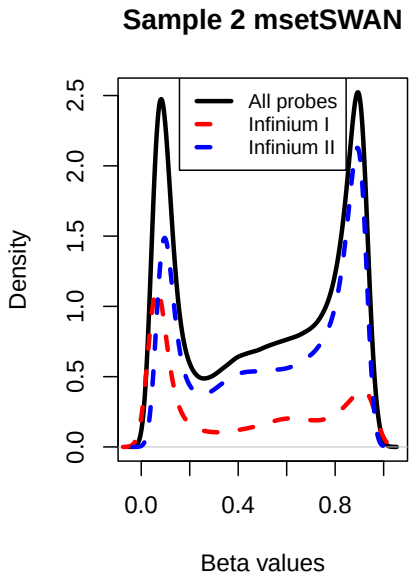
From all plots there are discrepancies in the grouping for sex and status.The plots between both methods for preprocessing are similar, while the plot with 10000 variable positions are noticeably different. However, the grouping in these plots still suggest that there may be errors in how the data was categorized.

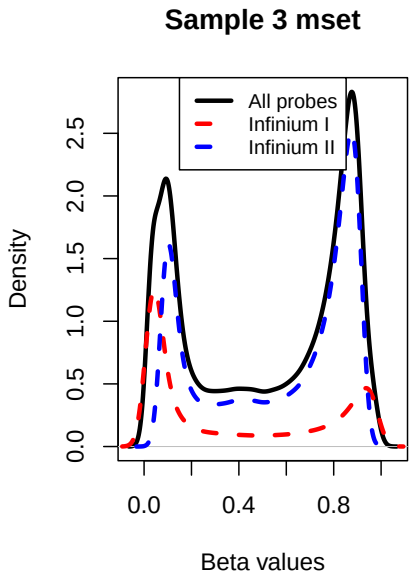
Part G

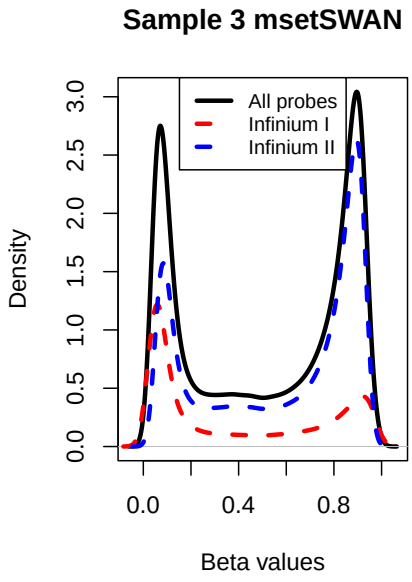
Plot the distribution of beta values before and after SWAN normalization using **plotBetasByType()**. What do you see in the density plots?

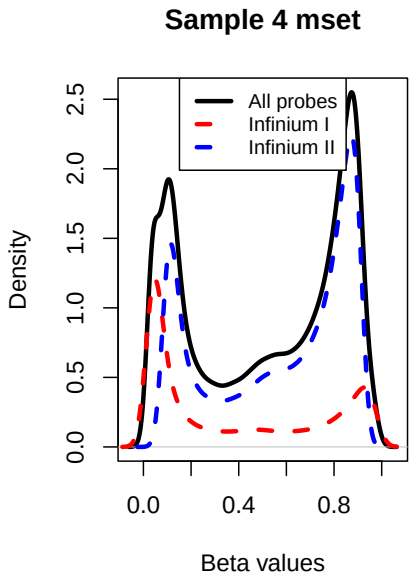


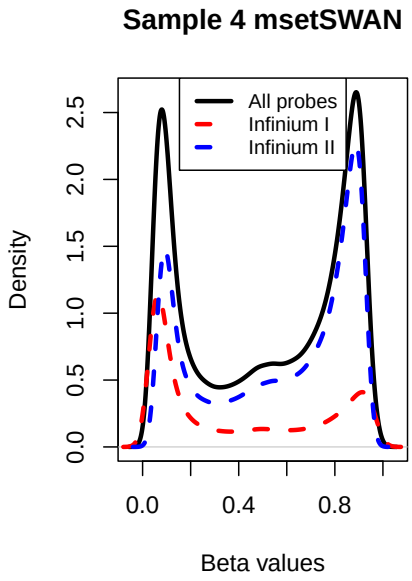

Sample 2 mset
All probes
Infinium I
Infinium II
Density
Beta values
0.0
0.5
1.0
1.5
2.0
2.5
0.0
0.4
0.8

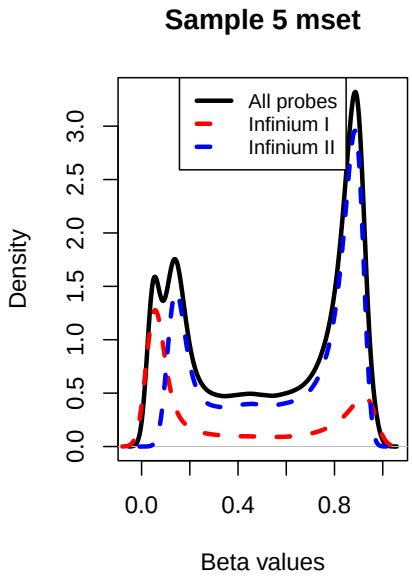

Sample 2 msetSWAN
All probes
Infinium I
Infinium II
Density
Beta values
0.0
0.5
1.0
1.5
2.0
2.5
0.0
0.4
0.8

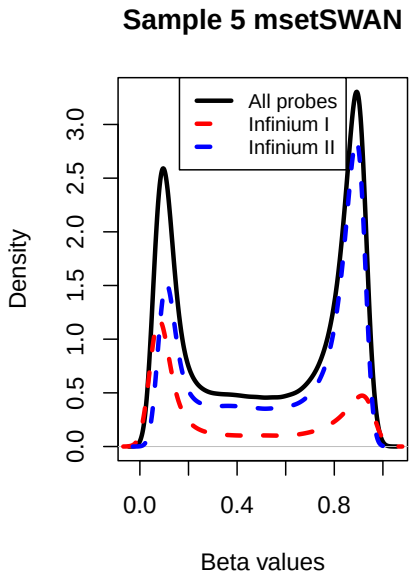

Sample 3 mset
All probes
Infinium I
Infinium II
Density
Beta values
0.0
0.5
1.0
1.5
2.0
2.5
0.0
0.4
0.8

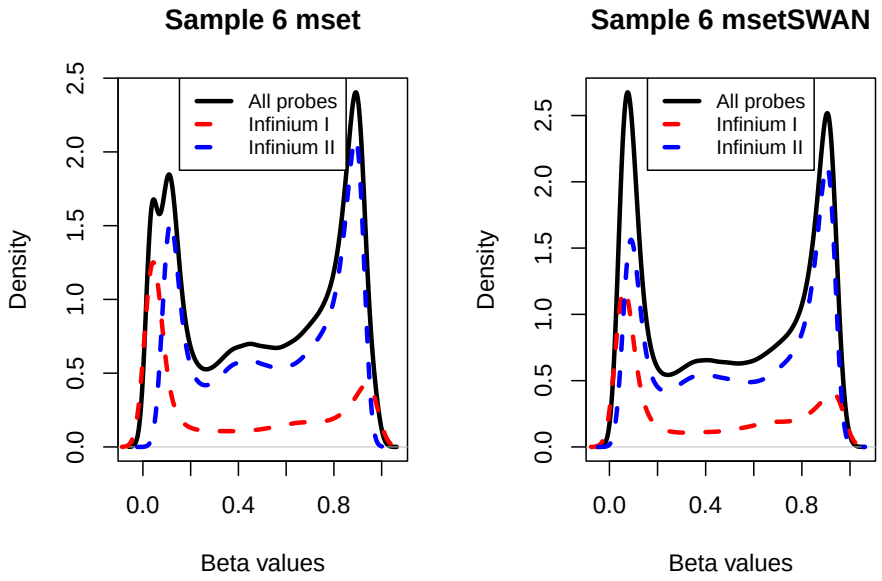

Sample 3 msetSWAN
All probes
Infinium I
Infinium II
Density
Beta values
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.0
0.4
0.8


Sample 4 mset
All probes
Infinium I
Infinium II
Density
0.0
0.5
1.0
1.5
2.0
2.5
0.0
0.4
0.8
Beta values


Sample 4 msetSWAN
All probes
Infinium I
Infinium II
Density
0.0
0.5
1.0
1.5
2.0
2.5
0.0
0.4
0.8
Beta values


Sample 5 mset
All probes
Infinium I
Infinium II
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.0
0.4
0.8
Beta values


Sample 5 msetSWAN
All probes
Infinium I
Infinium II
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.0
0.4
0.8
Beta values



The plots on the left show the results before SWAN normalization, and the plots on the right show the results after SWAN normalization. After normalization the the plots have much cleaner peaks, whereas before normalization the plots tend to have peaks on the left hand side of the plots which have multiple peaks at the apex of the all probes line.

Question 2: DNA Methylation Annotation and Differentially Methlyated Positions (Illumina 450K)

Part A

What are CpG islands, shores, shelves and open seas? From **annotation()** how many CpG site probes are in each of these types?

- **CpG Islands**: CpG islands are clusters of unmethylated CpGs occuring near highly expressed genes.
- **CpG Shores**: CpG shores are regions that exist on either side of a CpG island, generally around 2Kb.
- **CpG Shelves**: CpG shelves are regions that exist on either side of CpG shores, generally around 2kB.
- **Open Seas**: Open seas are regions that are more than 4kB away from CpG islands.

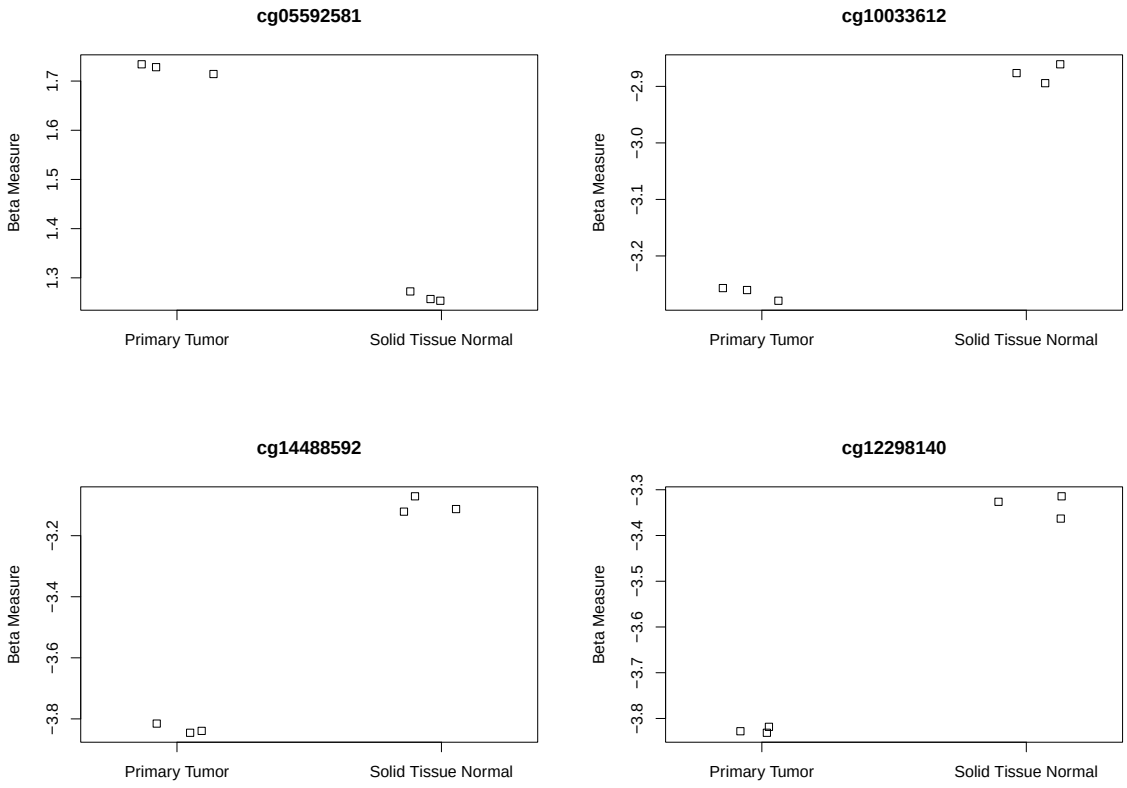
Table 3: Number of CpG Probes by Site

CpG Site	Number of Probes
CpG Islands	150254
CpG Upstream Shores	62870
CpG Downstream Shores	49197
CpG Upstream Shelves	24844
CpG Downstream Shelves	22300
CpG Open Sea	176047

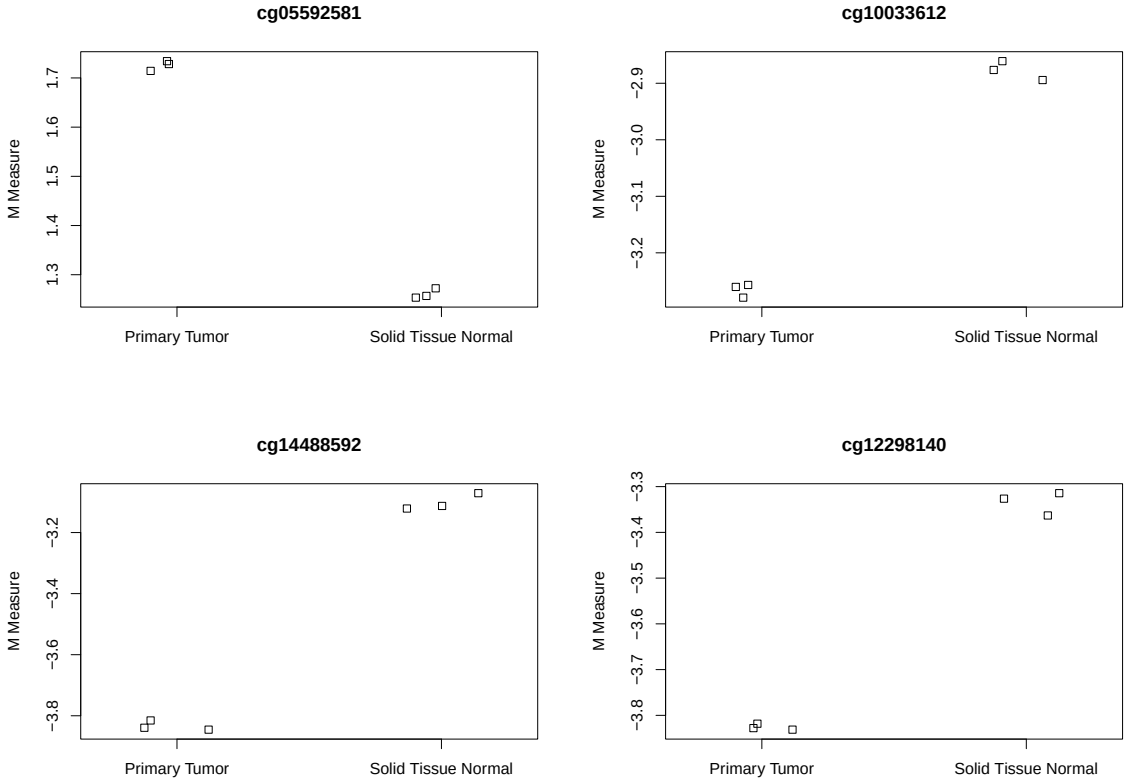
Part B

Using the SWAN normalized data from problem #1 **msetSWAN**, find differentially methylated positions (DMP) for cancer status with **getM()**, followed by **dmpFinder()** (which currently does not handle paired samples, so you will need to run it assuming independence). Are there any DMPs with q-value ≤ 0.10? Using a p-value cutoff of 10^{-5}, how many DMPs show hyper or hypomethylation due to cancer status? Use **plotCpg()** to plot the beta values and then M-values for the top four DMPs. What trends and effect sizes do you see in the plots?

There are 0 differentially methylated positions with q-values ≤ 0.10. There are 8 differentially methylated positions between cancer and non-cancer samples. Of these 8, 7 are hypomethylated and 1 are hypermethylated.



The beta measures are lower for the primary tumor samples for all differentially methylated regions except for cg05592581. Additionally, this sample is the only one where the beta measures are positive for each sample type instead of negative.

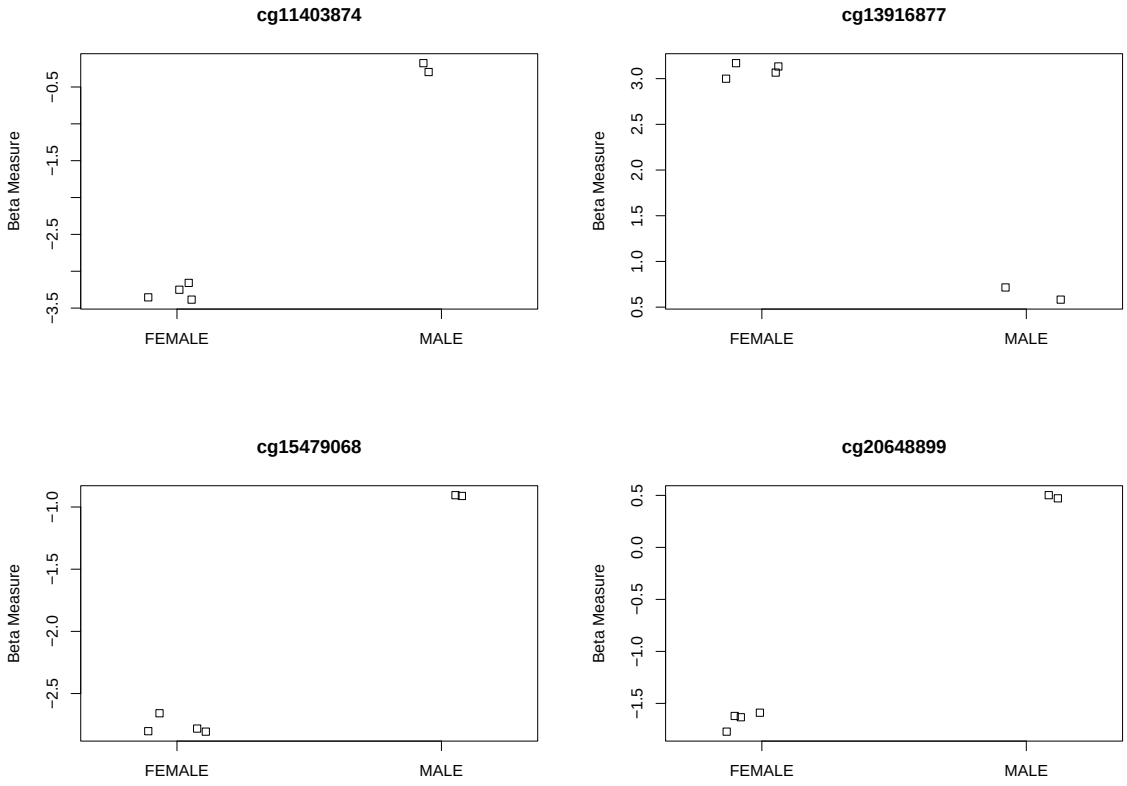


The M measures are lower for the primary tumor samples for all differentially methylated regions except for cg05592581. As with the beta values, this was the only sample where the M measures are positive for each sample type instead of negative. Between both sets of plots the general trends are the same.

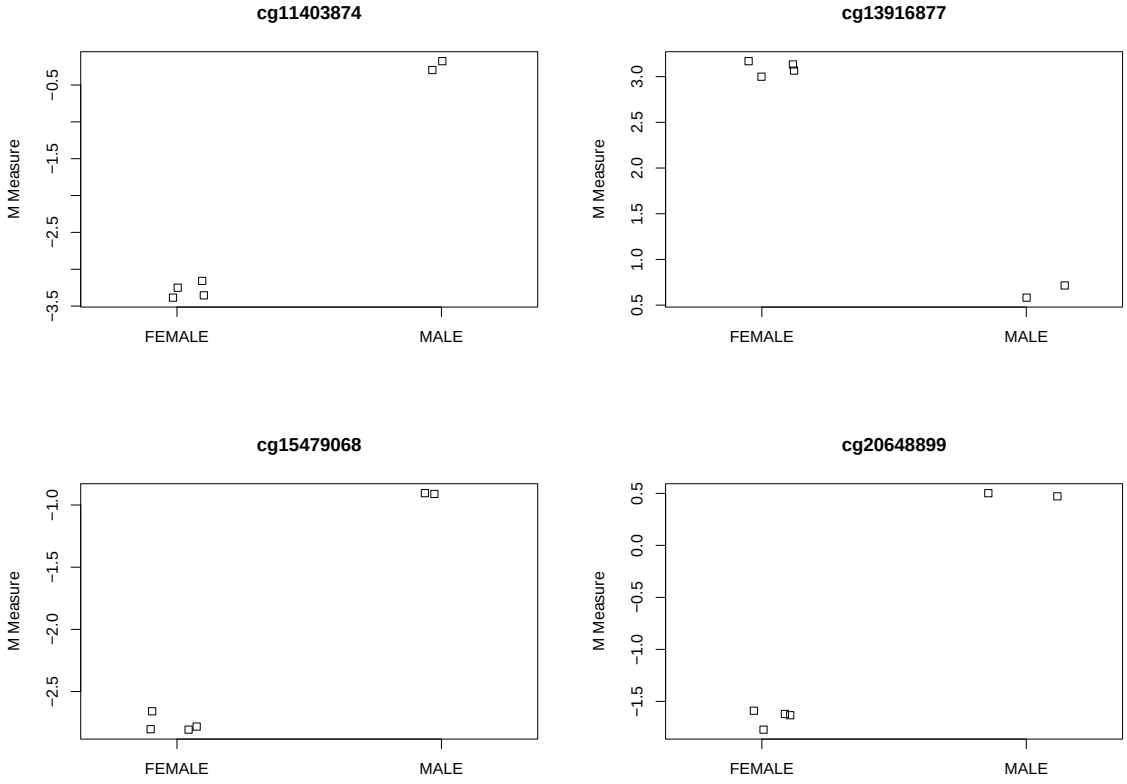
Part C

Repeat part b) but for DMPs between male and females.

There are 0 differentially methylated positions with q-values ≤ 0.10. There are 17 differentially methylated positions between males and females. Of these 17, 11 are hypomethylated and 6 are hypermethylated.



The beta measures are lower for female subjects for all differentially methylated regions with the exception of cg13916877. For all other samples males are higher. Samples from both sexes were above 0 for cg13916877, and for cg20648899 females have negative beta values and males have positive values.



The M measures are lower for female subjects for all differentially methylated regions with the exception of cg13916877, same as with the beta plots. Again, both sexes were above 0 for cg13916877, and for cg20648899 females have negative beta values and males have positive values. As in part b the trends between the two sets of plots are similar.

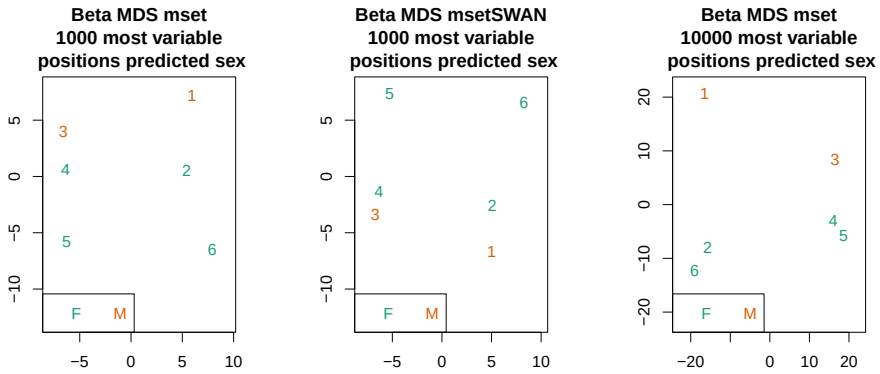
Part D

Global methylation profiles vary by sex. There is a function **addSex()** to estimate whether each samples is male or female. Are the predicted and given labels correct for Sex? If not, revisit the MDS plot from part 1e)? Do the new predictions group in the plot? Also repeat the analysis in 2c). Now are there DMPs with q-value ≤ 0.10 (or p-value $\leq 10^{-5}$)?

Table 4: Predicted vs. Reported Sex

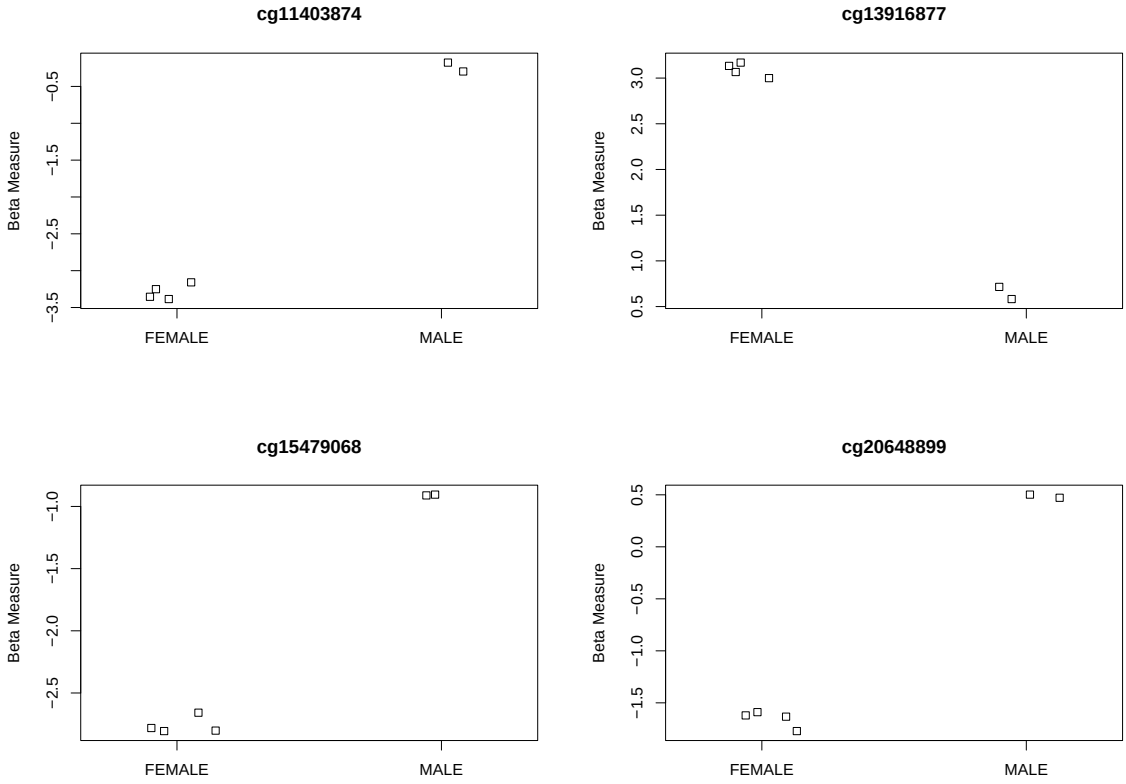
ID	Predicted	Reported
1	M	MALE
2	F	MALE
3	M	FEMALE
4	F	FEMALE
5	F	FEMALE
6	F	FEMALE

From the Table 4 above there are discrepancies between reported and predicted sex. Sample 2 was reported as male and predicted as female, and sample 3 was reported as male and predicted as female.

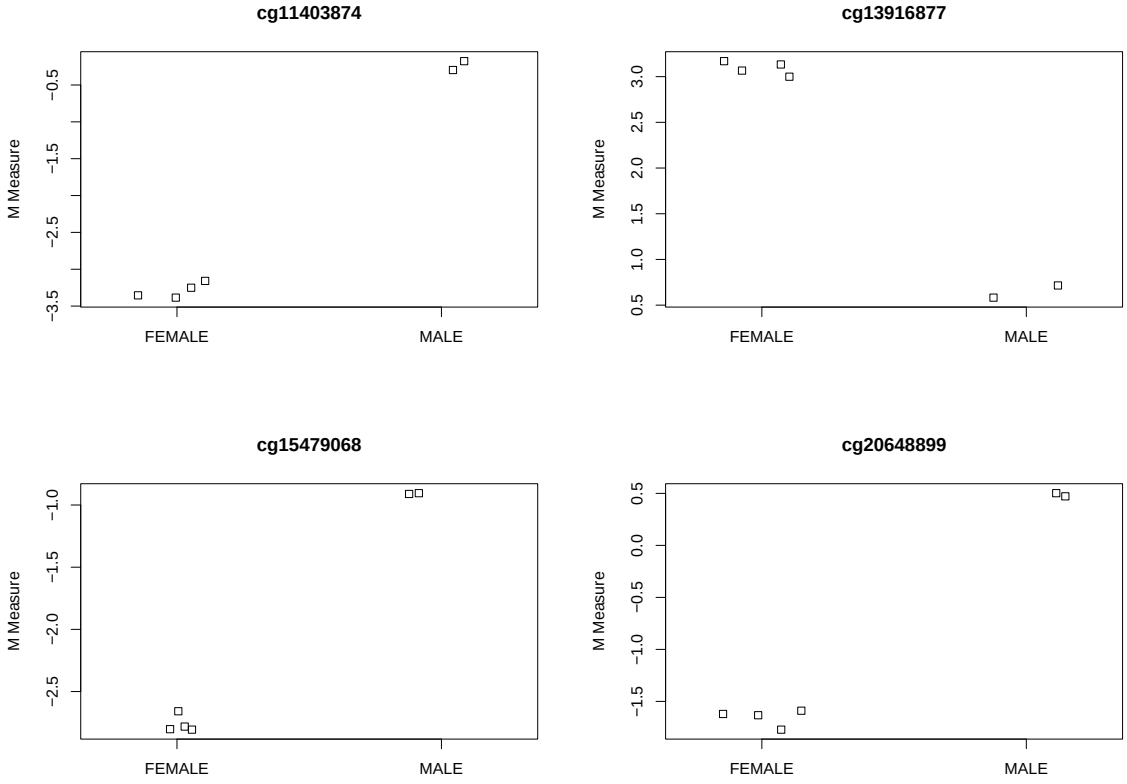


After remaking the plots from part 1e the predicted sexes group horizontally.

There are 0 differentially methylated positions with q-values ≤ 0.10. There are 17 differentially methylated positions between cancer and non-cancer samples. Of these 17, 7 are hypomethylated and 1 are hypermethylated.



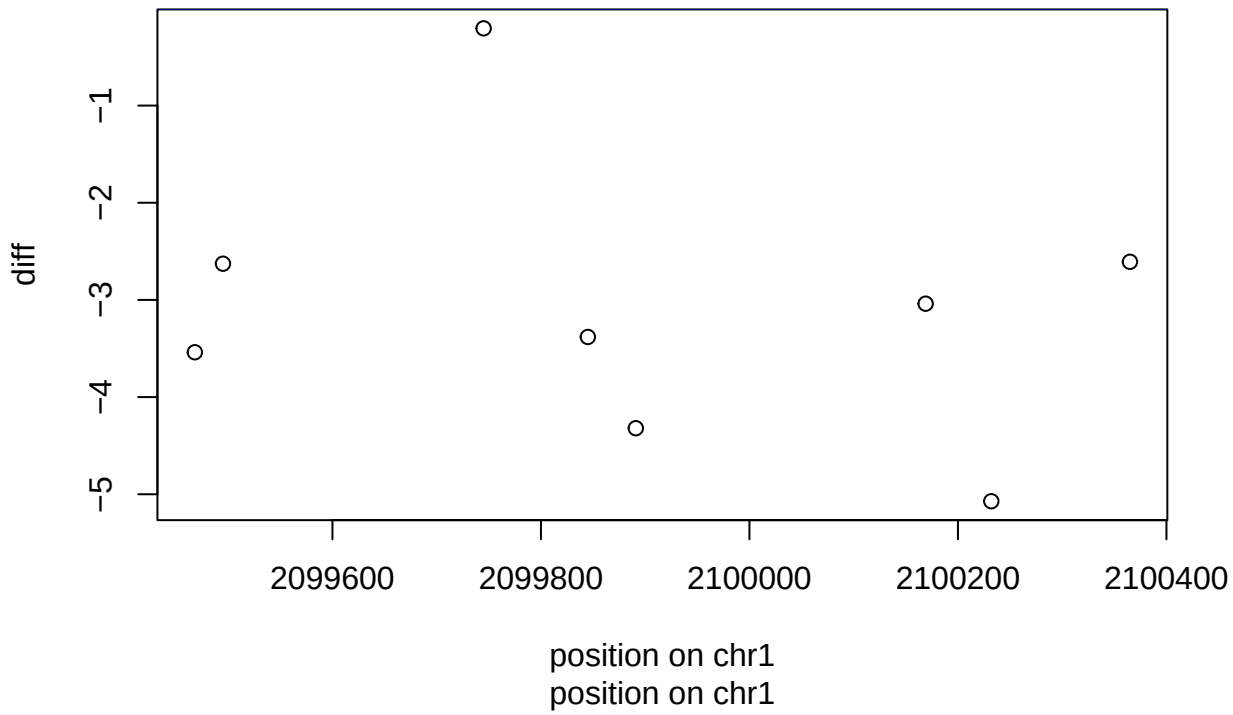
After plotting the top four most significant CpG sites there are new differentially methylated sites. Males tend to have higher beta measurements for all CpG sites, with the exception of cg13916877. This is also the only plot where both sexes are above 0. For cg20648899 males have a positive beta measurement and females have negative beta measurements.



The above plots show the results after plotting for M measurement. Males tend to have higher M measurements for all CpG sites, with the exception of cg13916877. This is also the only plot where both sexes are above 0. For cg20648899 males have a positive beta measurement and females have negative beta measurements.The trends are similar to those for the beta plots.

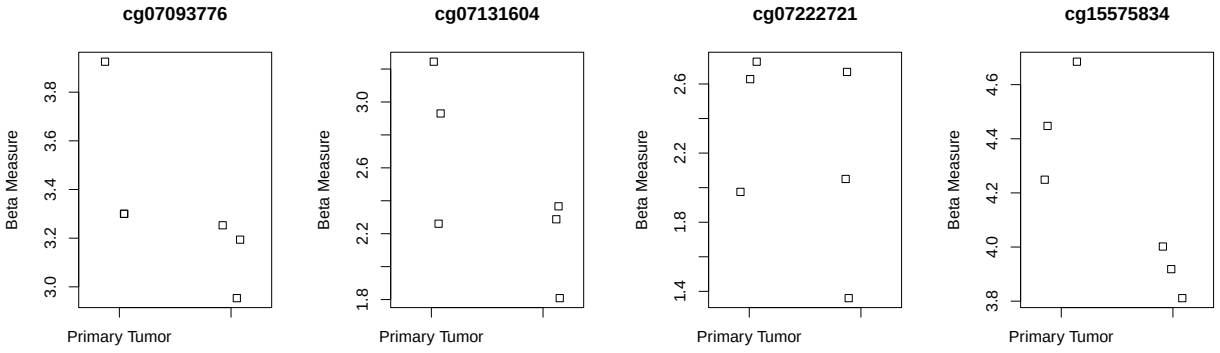
Part E

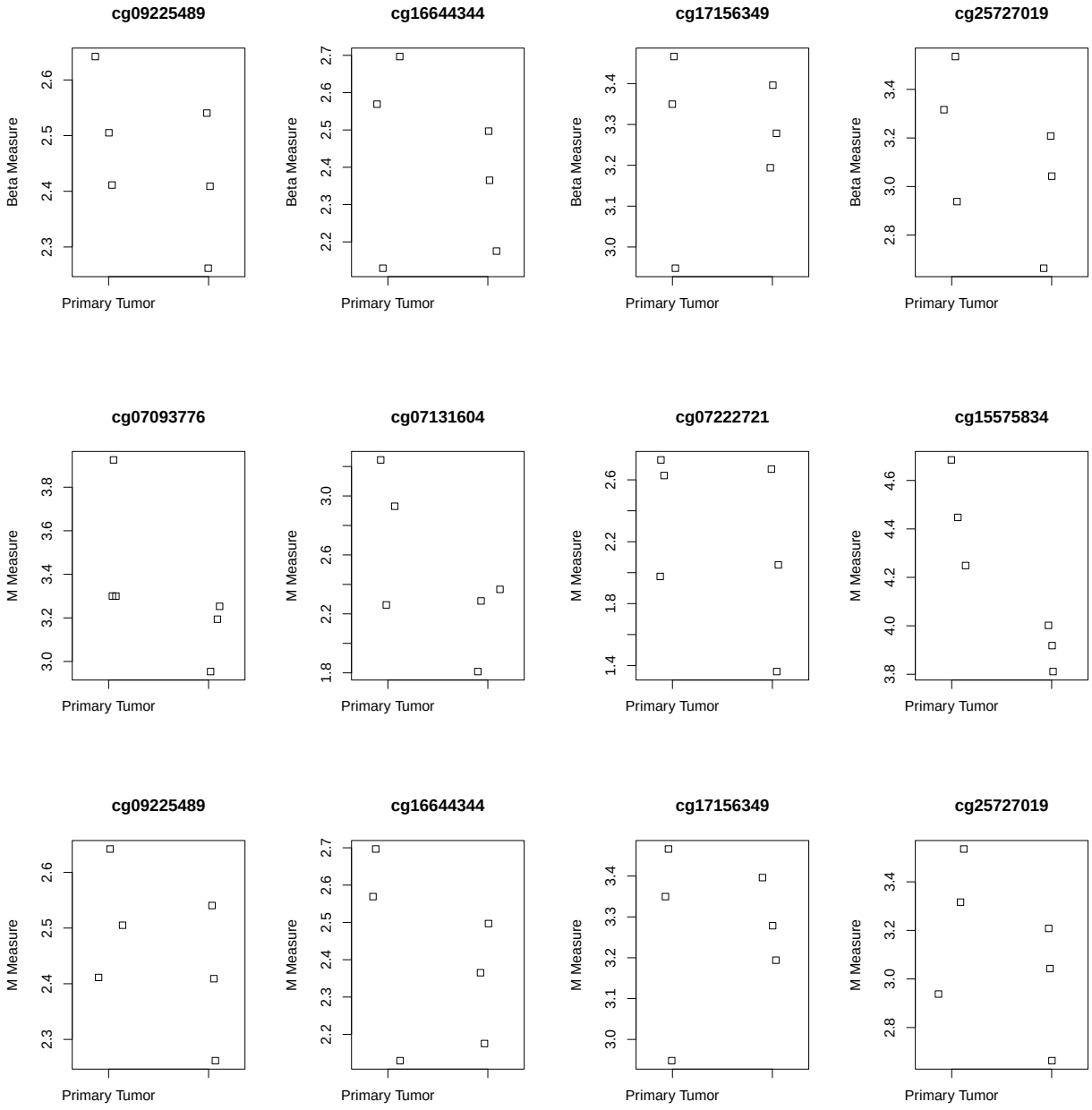
This samples data is too small for **bumphunter** to identify significant regions by performing permutations or bootstrap. However, we can use the **getSegment()** function to find region of extreme values for the differences found in part b) and use the following code to plot one examples region. Note, this is just an example. Report on, and provide a plot for a region that shows hypomethylation in more than one CpG site for the cancer subjects.



position on chr1

The plot above shows multiple hypomethylated CpG sites. The plots below show the beta and M plots for each methylation site after selecting it out from the DMP object.




cg09225489
cg16644344
cg17156349
cg25727019
Beta Measure
Primary Tumor
cg07093776
cg07131604
cg07222721
cg15575834
M Measure
Primary Tumor
cg09225489
cg16644344
cg17156349
cg25727019
M Measure
Primary Tumor

CODE

```
library(tidyverse)
library(shinyMethyl)
library(minfi)
library(bumphunter)
library(IlluminaHumanMethylation450kmanifest)
library(IlluminaHumanMethylation450kanno.ilmn12.hg19)
library(kableExtra)
library(latex2exp)


### START QUESTION 1 CODE ###

# Loading in data
baseDir <- c("C:/Users/domin/Documents/Biostatistics Masters Program/Spring 2024/SMG-BIOS7659/Homework 
targets <- read.metharray.sheet(baseDir)
rgSet <- read.metharray.exp(targets = targets)
annotation(rgSet)

# Saving methylation signals for raw data and the SWAN normalization method.
mset <- preprocessRaw(rgSet)
set.seed(303)
msetSWAN <- preprocessSWAN(rgSet)


## START QUESTION 1 PART A CODE ##

# Extracting demographic data.
demographic_data <- data.frame(pData(rgSet))

age_mean <- mean(demographic_data$patient.age_at_initial_pathologic_diagnosis)
age_sd <- sd(demographic_data$patient.age_at_initial_pathologic_diagnosis)

height_mean <- mean(demographic_data$patient.height)
height_sd <- sd(demographic_data$patient.height)

weight_mean <- mean(demographic_data$patient.weight)
weight_sd <- sd(demographic_data$patient.weight)

race_white <- nrow(demographic_data[
  demographic_data$patient.race == "WHITE",])/2

race_aa <- nrow(demographic_data[
  demographic_data$patient.race == "BLACK OR AFRICAN AMERICAN",])/2

status_cancer <- nrow(demographic_data[demographic_data$Status == "cancer",])
status_normal <- nrow(demographic_data[demographic_data$Status == "normal",])

sex_male <- nrow(demographic_data[demographic_data$Sex == "MALE",])/2
sex_female <- nrow(demographic_data[demographic_data$Sex == "FEMALE",])/2

age_table <- paste(round(age_mean,2),"(",round(age_sd,2),")",sep = "")
```

```r
height_table <- paste(round(height_mean,2),"(",round(height_sd,2),")",sep= "")
weight_table <- paste(round(weight_mean,2),"(",round(weight_sd,2),")",sep="")

demographic_table <- rbind(age_table,height_table,weight_table,
                           race_white,race_aa,status_cancer,
                           status_normal,sex_male,sex_female)


rownames(demographic_table) <- c("Age(mean,sd)","Height(mean,sd)",
                                 "Weight(mean,sd)","White",
                                 "African American","With Cancer",
                                 "Without Cancer","Male","Female")

table_1 <- kbl(demographic_table,
               caption = "Demographic Data",
               booktabs = T, align = "c") %>%
  kable_styling(latex_options = "HOLD_position") %>%
  group_rows("Patient Race",4,5) %>%
  group_rows("Sample Type (n = 6)",6,7) %>%
  group_rows("Patient Sex (n = 3)",8,9)

table_1

## FINISH QUESTION 1 PART A CODE ##



## START QUESTION 1 PART B CODE ##

# Determining the number of type 1 and 2 probes.
probe_info <- getManifest(rgSet)


## START QUESTION 1 PART C CODE ##

par(mfrow = c(1,2))

densityPlot(rgSet,sampGroups = pData(rgSet)$sample_type,main = "Sample Type",
            ylim = c(0,4.5))
densityBeanPlot(rgSet,sampGroups = pData(rgSet)$sample_type,
                sampNames = targets$id,main = "Sample Type - Beta")


par(mfrow = c(1,2))

densityPlot(rgSet,sampGroups = pData(rgSet)$Sex,main = "Sex",
            ylim = c(0,4.5))
densityBeanPlot(rgSet,sampGroups = pData(rgSet)$Sex,sampNames = targets$id,
                main = "Sex - Beta")

## FINISH QUESTION 1 PART C CODE ##
```

```r
## START QUESTION 1 PART D CODE ##

controlStripPlot(rgSet,controls = "BISULFITE CONVERSION I",
                 sampNames = pData(rgSet)$id)


controlStripPlot(rgSet,controls = "NEGATIVE",
                 sampNames = pData(rgSet)$id)

## FINISH QUESTION 1 PART D CODE ##


## START QUESTION 1 PART E CODE ##

# Calculating p-values.
pvalues <- data.frame(detectionP(rgSet))

# Summing up p-values >0 0.05 for each sample.
failed_samples <- data.frame(colSums(pvalues>0.05)/nrow(pvalues))
colnames(failed_samples) <- "Percentage p-values >= 0.05"
rownames(failed_samples) <- pData(rgSet)$id
failed_samples <- rownames_to_column(failed_samples,"Sample")

failed_samples_table <- kbl(failed_samples,
                      caption = "P-Values >= to 0.05",
                      booktabs = T, align = "lc",escape = F) %>%
  kable_styling(latex_options = "HOLD_position")

failed_samples_table


# Determining how many probes have average detection p-values >= 0.05
# across the 6 samples.

failed_rows <- data.frame(rowMeans(pvalues)) %>%
  mutate(rowMeans.pvalues. = rowMeans.pvalues. > 0.05 ) %>%
  colSums()

## FINISH QUESTION 1 PART E CODE ##


## START QUESTION 1 PART F CODE ##

# Plotting using mset data.
par(mfrow = c(1,2))

# Sample group by sex for mset data.
mdsPlot(mset,sampNames = mset$id,sampGroups = mset$Sex, ylim = c(-13,8),
        main = "Beta MDS mset\n 1000 most variable \n positions by sex")


# Sample group by cancer status for mset data.
mdsPlot(mset,sampNames = mset$id,sampGroups = mset$Status, ylim = c(-13,8),
```

```
        legendNCol = 1,
        main = "Beta MDS mset\n 1000 most variable \n positions by status")


par(mfrow = c(1,2))

# Sample group by sex for msetSWAN data.
mdsPlot(msetSWAN,sampNames = pData(msetSWAN)$id,
        sampGroups = pData(msetSWAN)$Sex, ylim = c(-13,8),
        main = "Beta MDS msetSWAN\n 1000 most variable \n positions by sex")

# Sample group by cancer status for msetSWAN data.
mdsPlot(msetSWAN,sampNames = pData(msetSWAN)$id,
        sampGroups = pData(msetSWAN)$Status, ylim = c(-13,8),legendNCol = 1,
        main = "Beta MDS msetSWAN\n 1000 most variable \n positions by Status")

## FINISH QUESTION 1 PART F CODE ##


par(mfrow = c(1,2))

# Sample group by sex for msetSWAN data.
mdsPlot(msetSWAN,sampNames = pData(msetSWAN)$id,
        sampGroups = pData(msetSWAN)$Sex,ylim = c(-25,25),
        numPositions = 10000,
        main = "Beta MDS msetSWAN\n 10000 most variable \n positions by sex")

# Sample group by cancer status for msetSWAN data.
mdsPlot(msetSWAN,sampNames = pData(msetSWAN)$id,
        sampGroups = pData(msetSWAN)$Status,legendNCol = 1,
        numPositions = 10000,ylim = c(-25,25),
        main = "Beta MDS msetSWAN\n 10000 most variable \n positions by Status")

## FINISH QUESTION 1 PART F CODE ##


## START QUESTION 1 PART G CODE ##

par(mfrow = c(1,2))

# Plotting Betas comparing before an after normalization for sample 1
plotBetasByType(mset[,1],main = "Sample 1 mset",cex.legend = 0.8)
plotBetasByType(msetSWAN[,1],main = "Sample 1 msetSWAN",
                cex.legend = 0.80)


par(mfrow = c(1,2))

# Plotting Betas comparing before an after normalization for sample 2
plotBetasByType(mset[,2],main = "Sample 2 mset",cex.legend = 0.8)
plotBetasByType(msetSWAN[,2],main = "Sample 2 msetSWAN",
                cex.legend = 0.80)
```

```r
par(mfrow = c(1,2))

# Plotting Betas comparing before an after normalization for sample 3
plotBetasByType(mset[,3],main = "Sample 3 mset",cex.legend = 0.8)
plotBetasByType(msetSWAN[,3],main = "Sample 3 msetSWAN",
                cex.legend = 0.80)


par(mfrow = c(1,2))

# Plotting Betas comparing before an after normalization for sample 4
plotBetasByType(mset[,4],main = "Sample 4 mset",cex.legend = 0.8)
plotBetasByType(msetSWAN[,4],main = "Sample 4 msetSWAN",
                cex.legend = 0.80)


par(mfrow = c(1,2))

# Plotting Betas comparing before an after normalization for sample 5
plotBetasByType(mset[,5],main = "Sample 5 mset",cex.legend = 0.8)
plotBetasByType(msetSWAN[,5],main = "Sample 5 msetSWAN",
                cex.legend = 0.80)


par(mfrow = c(1,2))

# Plotting Betas comparing before an after normalization for sample 6
plotBetasByType(mset[,6],main = "Sample 6 mset",cex.legend = 0.8)
plotBetasByType(msetSWAN[,6],main = "Sample 6 msetSWAN",
                cex.legend = 0.80)

## FINISH QUESTION 1 PART G CODE ##

### FINISH QUESTION 1 CODE ###


### START QUESTION 2 CODE ###

# Setting up data.
gset <- mapToGenome(msetSWAN)
annotation <- data.frame(getAnnotation(gset))


## START QUESTION 2 PART A CODE ##

# Finding number of CpG islands.
cpg_island_number <- nrow(annotation[
  annotation$Relation_to_Island == "Island",])

# Finding number of CpG upstream shores.
cpg_up_shores_number <- nrow(annotation[
  annotation$Relation_to_Island == "N_Shore",])
```

```r
# Finding number of CpG downstream shores.
cpg_down_shores_number <- nrow(annotation[
  annotation$Relation_to_Island == "S_Shore",])

# Finding number of upstream CpG shelves.
cpg_up_shelves <- nrow(annotation[
  annotation$Relation_to_Island == "N_Shelf",])

# Finding number of downstream CpG shelves.
cpg_down_shelves <- nrow(annotation[
  annotation$Relation_to_Island == "S_Shelf",])

# Finding number of CpG open sea sites.
cpg_opensea <- nrow(annotation[annotation$Relation_to_Island == "OpenSea",])

cpg_site_df <- rbind(cpg_island_number,cpg_up_shores_number,
                     cpg_down_shores_number,cpg_up_shelves,
                     cpg_down_shelves,cpg_opensea)

rownames(cpg_site_df) <- c("CpG Islands","CpG Upstream Shores",
                           "CpG Downstream Shores","CpG Upstream Shelves",
                           "CpG Downstream Shelves","CpG Open Sea")

cpg_site_df <- cpg_site_df %>% data.frame() %>% rownames_to_column()

colnames(cpg_site_df) <- c("CpG Site","Number of Probes")

cpg_site_table <- kbl(cpg_site_df,
                      caption = "Number of CpG Probes by Site",
                      booktabs = T, align = "lc") %>%
  kable_styling(latex_options = "HOLD_position")

cpg_site_table

## FINISH QUESTION 2 PART A CODE ##


## START QUESTION 2 PART B CODE ##

# Getting the  and identifying differentially methylated regions.
M_cancer <- getM(gset)
DMR_cancer <- data.frame(dmpFinder(M_cancer,
                                   pheno = pData(gset)$sample_type))

# Determining if there are any q-values <= 0.10.
q_value_number_cancer <- nrow(DMR_cancer[DMR_cancer$qval <= 0.10,])

# Determining number of hyper and hypomethylation positions due to cancer status
# for a p-value cutoff of 10e-5.
p_value_number_cancer <- DMR_cancer[DMR_cancer$pval < 1e-5,]
number_p_cancer <- nrow(p_value_number_cancer)

hypo_cancer <- nrow(p_value_number_cancer[p_value_number_cancer$intercept < 0,])
```

```r
hyper_cancer <- nrow(p_value_number_cancer[
  p_value_number_cancer$intercept > 0,])

# Sorting for most significant p-values and selecting top four.
p_value_sig_cancer <- head(p_value_number_cancer[order(
  p_value_number_cancer$pval),],4)

# Selecting top 4 p-values from M
mset_cancer <- M_cancer[c("cg05592581","cg14488592","cg10033612","cg12298140"),]


par(mfrow = c(2,1))

# Plotting beta values for cancer groups.
plotCpg(mset_cancer,cpg = c("cg05592581","cg14488592",
                            "cg10033612","cg12298140"),
        pheno = pData(gset)$sample_type, measure = "beta",
        type = "categorical", ylab = "Beta Measure",xlab = "Sample Type")



par(mfrow = c(2,1))

# Plotting M values for cancer groups.
plotCpg(mset_cancer,cpg = c("cg05592581","cg14488592",
                            "cg10033612","cg12298140"),
        pheno = pData(gset)$sample_type, measure = "M",
        type = "categorical", ylab = "M Measure",xlab = "Sample Type")

## FINISH QUESTION 2 PART B CODE ##

## START QUESTION 2 PART C CODE ##

# Getting the  and identifying differentially methylated regions.
M_sex <- getM(gset)
DMR_sex <- data.frame(dmpFinder(M_sex,
                                pheno = pData(gset)$Sex))

# Determining if there are any q-values <= 0.10.
q_value_number_sex <- nrow(DMR_sex[DMR_sex$qval <= 0.10,])

# Determining number of hyper and hypomethylation positions due to sex
# for a p-value cutoff of 10e-5.
p_value_number_sex <- DMR_sex[DMR_sex$pval < 1e-5,]
number_p_sex <- nrow(p_value_number_sex)

hypo_sex <- nrow(p_value_number_sex[p_value_number_sex$intercept < 0,])
hyper_sex <- nrow(p_value_number_sex[
  p_value_number_sex$intercept > 0,])

# Sorting for most significant p-values and selecting top four.
p_value_sig_sex <- head(p_value_number_sex[order(
  p_value_number_sex$pval),],4)
```

```r
# Selecting top 4 p-values from M
mset_sex <- M_sex[c("cg11403874","cg15479068","cg13916877","cg20648899"),]


par(mfrow = c(2,1))

# Plotting beta values for males and females.
plotCpg(mset_sex,cpg = c("cg11403874","cg15479068",
                         "cg13916877","cg20648899"),
        pheno = pData(gset)$Sex, measure = "beta",
        type = "categorical", ylab = "Beta Measure", xlab = "Sex")



par(mfrow = c(2,1))

# Plotting M values for cancer groups.
plotCpg(mset_sex,cpg = c("cg11403874","cg15479068",
                         "cg13916877","cg20648899"),
        pheno = pData(gset)$Sex, measure = "M",
        type = "categorical", ylab = "M Measure",xlab = "Sex")

## FINISH QUESTION 2 PART C CODE ##

## START QUESTION 2 PART D CODE ##

gset_sex <- addSex(gset)
results_gset_sex <- cbind(pData(gset_sex)$id,pData(gset_sex)$predictedSex,
                          pData(gset_sex)$Sex)

gset_sex_table <- kbl(results_gset_sex,
                      caption = "Predicted vs. Reported Sex",
                      col.names = c("ID","Predicted","Reported"),
                      booktabs = T, align = "cll") %>%
  kable_styling(latex_options = "HOLD_position")

gset_sex_table


# Adding in predicted sex.
rgSet_new <- rgSet
rgSet_new$predictedSex <- pData(gset_sex)$predictedSex
mset_new <- preprocessRaw(rgSet_new)
msetSWAN_new <- preprocessSWAN(rgSet_new)

# Plotting using mset data.
par(mfrow = c(1,2))

# Sample group by predicted sex.
mdsPlot(mset_new,sampNames = mset_new$id,sampGroups = mset_new$predictedSex,
        ylim = c(-13,8),
        main = "Beta MDS mset\n 1000 most variable \n positions predicted sex")
```

```r
mdsPlot(msetSWAN_new,sampNames = msetSWAN_new$id,
        sampGroups = msetSWAN_new$predictedSex, ylim = c(-13,8),
        main = "Beta MDS msetSWAN\n 1000 most variable \n positions predicted sex")

mdsPlot(mset_new,sampNames = mset_new$id,sampGroups = mset_new$predictedSex,
        ylim = c(-22,22),
        numPositions = 10000,
        main = "Beta MDS mset\n 10000 most variable \n positions predicted sex")


# Repeating part C with the new data.
# Getting the  and identifying differentially methylated regions.
M_sex2 <- getM(gset_sex)
DMR_sex2 <- data.frame(dmpFinder(M_sex2,
                                 pheno = pData(gset_sex)$Sex))

# Determining if there are any q-values <= 0.10.
q_value_number_sex2 <- nrow(DMR_sex2[DMR_sex2$qval <= 0.10,])

# Determining number of hyper and hypomethylation positions due to sex
# for a p-value cutoff of 10e-5.
p_value_number_sex2 <- DMR_sex2[DMR_sex2$pval < 1e-5,]
number_p_value <- nrow(p_value_number_sex2)

hypo_sex2 <- nrow(p_value_number_sex2[p_value_number_sex2$intercept < 0,])
hyper_sex2 <- nrow(p_value_number_sex2[
  p_value_number_sex$intercept > 0,])

# Sorting for most significant p-values and selecting top four.
p_value_sig_sex2 <- head(p_value_number_sex2[order(
  p_value_number_sex2$pval),],4)

# Selecting top 4 p-values from M
mset_sex2 <- M_sex2[c("cg11403874","cg15479068","cg13916877","cg20648899"),]


par(mfrow = c(2,1))

# Plotting beta values for males and females.
plotCpg(mset_sex2,cpg = c("cg11403874","cg15479068",
                          "cg13916877","cg20648899"),
        pheno = pData(gset_sex)$Sex, measure = "beta",
        type = "categorical", ylab = "Beta Measure", xlab = "Sex")



par(mfrow = c(2,1))

# Plotting M values for cancer groups.
plotCpg(mset_sex2,cpg = c("cg11403874","cg15479068",
                          "cg13916877","cg20648899"),
        pheno = pData(gset_sex)$Sex, measure = "M",
        type = "categorical", ylab = "M Measure",xlab = "Sex")
```

```
## FINISH QUESTION 2 PART D CODE ##

## START QUESTION 2 PART E CODE ##

diffs <- DMR_cancer$intercept
chr <- annotation$chr
pos <- annotation$pos
cl <- clusterMaker(chr,pos,maxGap = 300) # cluster probes.

# Find regions with a stretch of differences.
# Setting 0 as cutoff to find hypomethylated regions.
segs <- getSegments(diffs, f= cl, cutoff = 0)

# Plotting a region with multiple hypomethylation sites.
j = 861
ind = segs$dnIndex[[j]]
index <- which(cl == cl[ind])
plot(pos[ind],diffs[ind],xlab = paste("position on", chr[ind]),
     ylab = "diff")
abline(h = 0, col = "blue")


# Selecting genes from annotation object.
annotated_subset <- annotation[c(ind),]

# Selecting out rows from DMP_cancer object.
DMP_cancer_subset <- DMR_cancer[rownames(annotated_subset),]

par(mfrow = c(1,2))

# Plotting beta values for cancer groups.
plotCpg(M_cancer[rownames(DMP_cancer_subset),],cpg = rownames(DMP_cancer_subset),
        pheno = pData(gset)$sample_type, measure = "beta",
        type = "categorical", ylab = "Beta Measure",xlab = "Sample Type")


par(mfrow = c(1,2))

# Plotting beta values for cancer groups.
plotCpg(M_cancer[rownames(DMP_cancer_subset),],cpg = rownames(DMP_cancer_subset),
        pheno = pData(gset)$sample_type, measure = "M",
        type = "categorical", ylab = "M Measure",xlab = "Sample Type")

## FINISH QUESTION 2 PART E CODE ##

### FINISH QUESTION 2 CODE ###
```