DNA/RNA DYNAMICS REPORT

Becchi Tommaso
University of Bologna, Master Degree Course in Bioinformatics

1) Load raw data with minfi and create an object called RGset storing the RGChannelSet object

- suppressMessages(library(minfi))
- SampleSheet <- read.table("Samplesheet_report_2020.csv",sep=",",header=T)
- targets <- read.metharray.sheet(getwd())

 # Read.metharray.sheet reads an Illumina methylation sample sheet, containing pheno-data information for the samples in an experiment.

- RGset <- read.metharray.exp(targets = targets)

 # Read.metharray.exp reads an entire methylation array experiment using a sample sheet or a target like data.frame.

- save(RGset,file="RGset.RData")

```
> RGset
class: RGChannelSet
dim: 622399 8
metadata(0):
assays(2): Green Red
rownames(622399): 10600313 10600322 ... 74810490 74810492
rowData names(0):
colnames(8): 5775278051_R01C01 5775278051_R04C02 ... 5930514035_R04C02
  5930514035_R06C02
colData names(7): Sample_Name Group ... Basename filenames
Annotation
  array: IlluminaHumanMethylation450k
  annotation: ilmn12.hg19
```

2) Create the dataframes Red and Green to store the red and green fluorescences respectively

- Red <- data.frame(getRed(RGset))

 # Gets the Red channel as a matrix.

- Green <- data.frame(getGreen(RGset))

 # Gets the Green channel as a matrix.

 # These functions belong to RGChannelSet-Class which allows to represent raw data from a two colors micro array

```
> head(Red)
         X5775278051_R01C01 X5775278051_R04C02 X5775278078_R02C01 X5775278078_R05C01
10600313                816               1055                603                647
10600322               2269               2965               1840               2103
10600328               1673               2500               1420               1588
10600336              18318              21740              16924              17981
10600345               4141               4575               3198               3123
10600353               1433               1680               1228               1211
         X5775278078_R05C02 X5930514034_R01C02 X5930514035_R04C02 X5930514035_R06C02
10600313                583                441                463                606
10600322               2228               1383               1823               2315
10600328               1600               3456               2773               2540
10600336              16158              16169              18221              19048
10600345               3233               3424               3336               3595
10600353               1141               1164                975               1136
```

```
> head(Green)
         X5775278051_R01C01 X5775278051_R04C02 X5775278078_R02C01 X5775278078_R05C01
10600313                373                494                278                360
10600322               7413              10620               6667               8362
10600328               2369               2375               1995               2179
10600336               1734               2060               1456               1775
10600345               3400               4416               3590               3665
10600353               4233               4873               3790               3406
         X5775278078_R05C02 X5930514034_R01C02 X5930514035_R04C02 X5930514035_R06C02
10600313                321                210                198                350
10600322               7963               6896               8344               9709
10600328               2401               2954               3222               2806
10600336               1757               1152               1702               1617
10600345               3559               2975               3516               4530
10600353               3693               2973               3371               3983
```

3) What are the Red and Green fluorescences for the address 71773431?

- Red['71773431',]
- Green['71773431',]

 # For each sample it is possible to extract the fluorescences emitted by the probe with a specific address.

- load("Illumina450Manifest.RData")
- Illumina450Manifest[Illumina450Manifest$AddressA_ID=="71773431",]

 # The address corresponds to a Type II probe

```
> Illumina450Manifest[Illumina450Manifest$AddressA_ID=="71773431",]
       IlmnID       Name AddressA_ID                                    AlleleA_ProbeSeq
31 cg01073572 cg01073572    71773431 CCCCTTAATTATCTAAAACCACAAATATACAACRAATAACCTCCTTCATC
   AddressB_ID AlleleB_ProbeSeq Infinium_Design_Type Next_Base Color_Channel
31                                                II

                  Forward_Sequence
31 ACTTAAAAGCCAAAATTTCATCAGCTAACAGGCTTAGAGTATGGAGCCTAAAGGACCTGT[CG]ATGAAGGAGGTTATTCGTTGCATATC
TGTGGTTTTAGACAATCAAGGGGGTTGTTTTTTT
   Genome_Build CHR  MAPINFO                                              SourceSeq
31           37   Y 16924422 CCCTTGATTGTCTAAAACCACAGATATGCAACGAATAACCTCCTTCATCG
   Chromosome_36 Coordinate_36 Strand Probe_SNPs Probe_SNPs_10 Random_Loci Methyl27_Loci
31             Y      15433816      F                                TRUE            NA
      UCSC_RefGene_Name        UCSC_RefGene_Accession UCSC_RefGene_Group
31 NLGN4Y;NLGN4Y;NLGN4Y NR_028318;NM_014893;NR_028319     Body;Body;Body
   UCSC_CpG_Islands_Name Relation_to_UCSC_CpG_Island Phantom DMR Enhancer HMM_Island
31                                                                 NA
   Regulatory_Feature_Name Regulatory_Feature_Group DHS
31                                              NA
```

Sample	Red fluorescence	Green fluorescence	Type	Color
X5775278051_R01C01	1052	3822	II	
X5775278051_R04C02	1400	4204	II	
X5775278078_R02C01	614	3070	II	
X5775278078_R05C01	918	3535	II	
X5775278078_R05C02	668	3639	II	
X5930514034_R01C02	579	3456	II	
X5930514035_R04C02	630	3644	II	
X5930514035_R06C02	751	4459	II	

4) Create the object MSet.raw

- MSet.raw <- preprocessRaw(RGset)

 # Preprocessing takes as input a RGChannelSet and returns a MethylSet. preprocessRaw converts the Red/Green channel for an Illumina methylation array into methylation signal, without using any normalization.

```
> MSet.raw
class: MethylSet
dim: 485512 8
metadata(0):
assays(2): Meth Unmeth
rownames(485512): cg00050873 cg00212031 ... ch.22.47579720R ch.22.48274842R
rowData names(0):
colnames(8): 5775278051_R01C01 5775278051_R04C02 ... 5930514035_R04C02
  5930514035_R06C02
colData names(7): Sample_Name Group ... Basename filenames
Annotation
  array: IlluminaHumanMethylation450k
  annotation: ilmn12.hg19
Preprocessing
  Method: Raw (no normalization or bg correction)
  minfi version: 1.34.0
  Manifest version: 0.4.0
```

5) Perform the following quality checks:

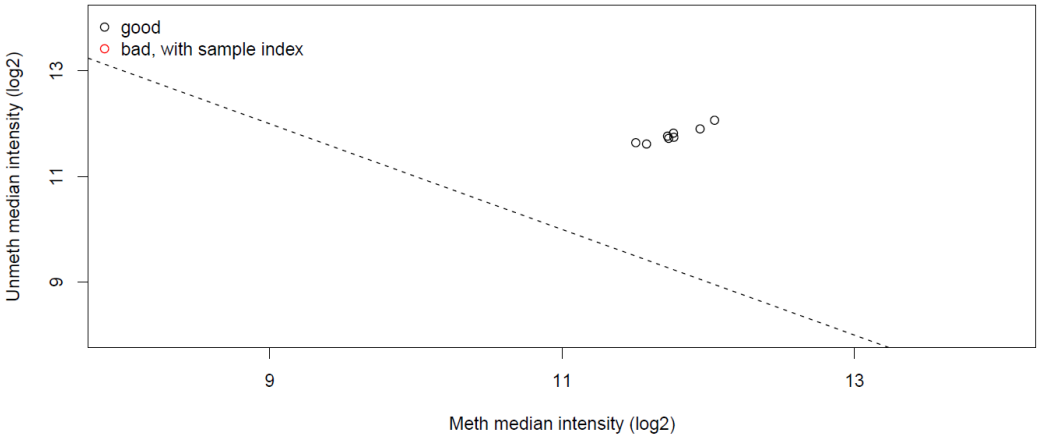
QC Plot

- qc <- getQC(MSet.raw)

#Estimate sample-specific quality control for methylation data.

```
> qc
DataFrame with 8 rows and 2 columns
                     mMed      uMed
                <numeric> <numeric>
5775278051_R01C01   11.7616   11.8222
5775278051_R04C02   12.0427   12.0668
5775278078_R02C01   11.5774   11.6170
5775278078_R05C01   11.7645   11.7444
5775278078_R05C02   11.7288   11.7241
5930514034_R01C02   11.5038   11.6416
5930514035_R04C02   11.7211   11.7661
5930514035_R06C02   11.9436   11.9035
```

- plotQC(qc)



 # It is possible to observe that all the data have good quality. The relationship between methylated intensity and unmethylated intensity is similar for different samples.

Intensity of negative controls

- getProbeInfo(RGset, type = "Control")
- df_TypeControl <- data.frame(getProbeInfo(RGset, type = "Control"))

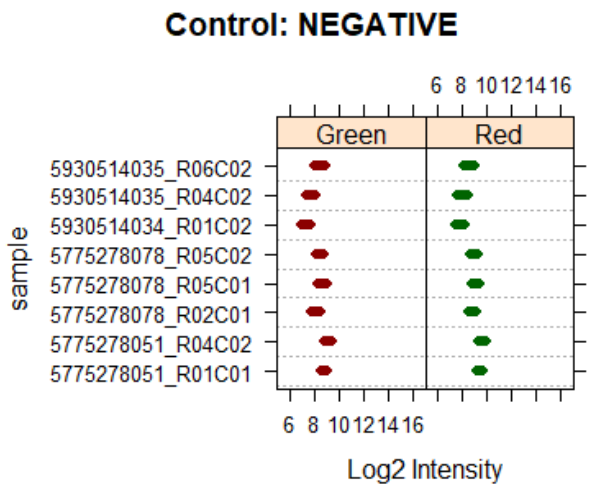
```
> table(df_TypeControl$Type)

 BISULFITE CONVERSION I BISULFITE CONVERSION II              EXTENSION
                     12                       4                       4
          HYBRIDIZATION                NEGATIVE         NON-POLYMORPHIC
                      3                     613                       4
                 NORM_A                  NORM_C                  NORM_G
                     32                      61                      32
                 NORM_T              RESTORATION           SPECIFICITY I
                     61                       1                      12
         SPECIFICITY II                STAINING          TARGET REMOVAL
                      3                       4                       2
```

 # Analysing the control probes, it is possible to observe that there are many different control types. Now we focus our attention on the negative control probes

- controlStripPlot(RGset, controls="NEGATIVE")

 # This function produces the control probe signal plot component of the QC report. "NEGATIVE" is selected as control type.



 # In this plot the colors are inverted because in the last version of R there is a problem with this package.
 # For each sample, the Negative Control signal is below 2^10=1024. This is positive because under normal experimental conditions, the signal intensities from negative controls can vary from approximately 100 to 1000 units.

Calculate detection pValues: for each sample, how many probes have a detection p-value higher than 0.05?

- detP <- detectionP(RGset)

 # A detection p-value is returned for every genomic position in every sample. Small p-values indicate a good position. Positions with non-significant p-values (typically >0.01) should not be trusted.

- save(detP,file="detP.RData")
- load("detP.RData")
- failed <- detP>0.05

 # For each sample it is possible to obtain the number of positions with a p-value higher than the threshold

- summary(failed)

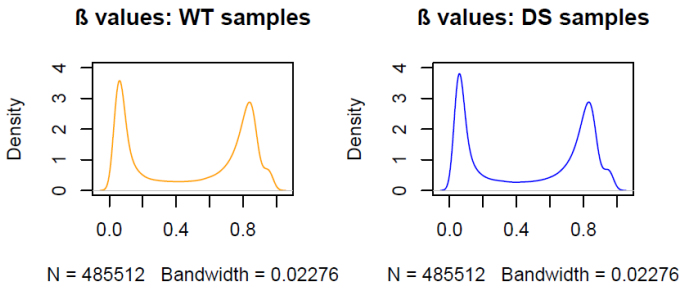
Sample	Failed positions
X5775278051_R01C01	247
X5775278051_R04C02	210
X5775278078_R02C01	264
X5775278078_R05C01	413
X5775278078_R05C02	385
X5930514034_R01C02	91
X5930514035_R04C02	46
X5930514035_R06C02	115

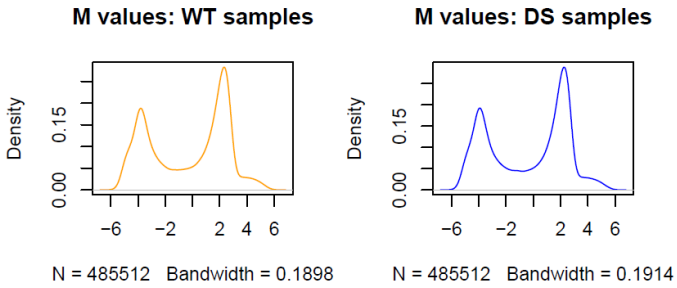
6) Calculate raw β and M values and plot the densities of mean methylation values, dividing the samples in DS and WT

- WT<-SampleSheet[SampleSheet$Group=="WT","Basename"]
- DS<-SampleSheet[SampleSheet$Group=="DS","Basename"]

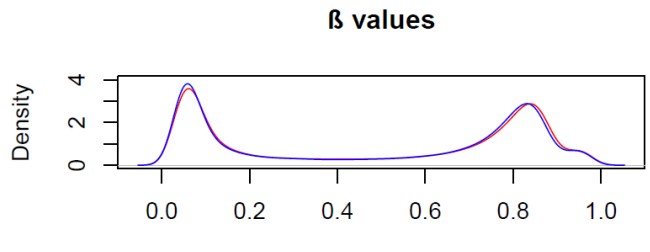
β= Meth / (Meth + Unmeth + offset)	*M = log(Beta) = log(Meth / Unmeth)*
beta <- getBeta(MSet.raw)	M <- getM(MSet.raw)
beta_WT<- beta[,colnames(beta)%in%WT]	M_WT<- M[,colnames(M)%in%WT]
beta_DS<- beta[,colnames(beta)%in%DS]	M_DS<- M[,colnames(M)%in%DS]
mean_of_beta_WT <- apply(beta_WT,1,mean,na.rm=T)	mean_of_M_WT <- apply(M_WT,1,mean,na.rm=T)
d_mean_of_beta_WT <- density(mean_of_beta_WT)	d_mean_of_M_WT <- density(mean_of_M_WT)
mean_of_beta_DS <- apply(beta_DS,1,mean,na.rm=T)	mean_of_M_DS <- apply(M_DS,1,mean,na.rm=T)
d_mean_of_beta_DS <- density(mean_of_beta_DS)	d_mean_of_M_DS <- density(mean_of_M_DS)

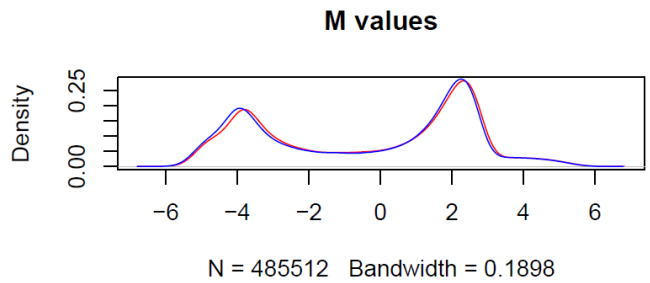
- par(mfrow=c(2,2))
- plot(d_mean_of_beta_WT,main="Density of Beta Values for WT samples",col="orange",ylim=c(0,4))
- plot(d_mean_of_beta_DS,main="Density of Beta Values for DS samples",col="blue",ylim=c(0,4))
- plot(d_mean_of_M_WT,main="Density of M Values for WT samples",col="orange")
- plot(d_mean_of_M_DS,main="Density of M Values for DS samples",col="blue")





- par(mfrow=c(2,1))
- plot(d_mean_of_beta_WT,main="β values",col="red",ylim=c(0,4))
- lines(d_mean_of_beta_DS, col="blue")
- plot(d_mean_of_M_WT,main="M values",col="red")
- lines(d_mean_of_M_DS, col="blue")





 # Observing the separate graphs and then superimposing the distributions for DS and WT it is observed that the results from the 2 different groups are very similar.

7) Normalize the data using SWAN normalization, compare raw data and normalized data.

- dfI <- Illumina450Manifest_clean[Illumina450Manifest_clean$Infinium_Design_Type=="I",]
- dfI <- droplevels(dfI)
- dfII <- Illumina450Manifest_clean[Illumina450Manifest_clean$Infinium_Design_Type=="II",]
- dfII <- droplevels(dfII)

 # RAW DATA

- beta_I <- beta[rownames(beta) %in% dfI$IlmnID,]
- beta_II <- beta[rownames(beta) %in% dfII$IlmnID,]
- mean_of_beta_I <- apply(beta_I,1,mean)
- mean_of_beta_II <- apply(beta_II,1,mean)
- d_mean_of_beta_I <- density(mean_of_beta_I,na.rm=T)
- d_mean_of_beta_II <- density(mean_of_beta_II,na.rm=T)
- sd_of_beta_I <- apply(beta_I,1,sd,na.rm=T)
- sd_of_beta_II <- apply(beta_II,1,sd,na.rm=T)
- d_sd_of_beta_I <- density(sd_of_beta_I,)
- d_sd_of_beta_II <- density(sd_of_beta_II)

 # SWAN NORMALIZED DATA

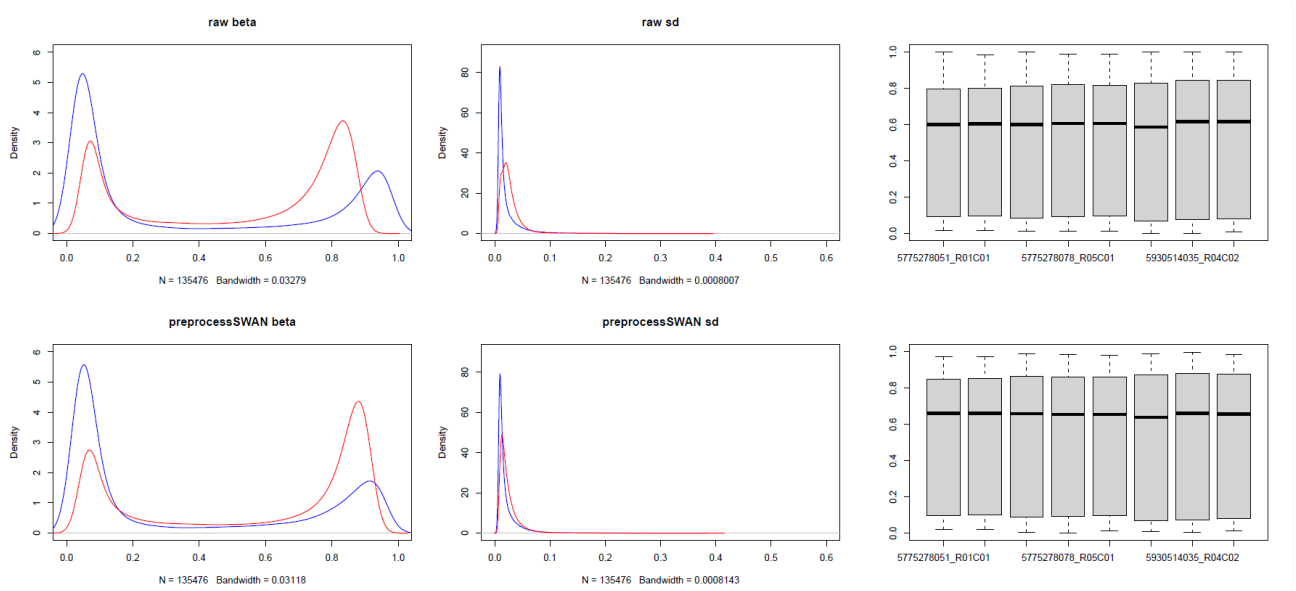
- preprocessSwan_results <- preprocessSWAN(RGset)

 # Subset-quantile Within Array Normalisation (SWAN) is a within array normalisation method for the Illumina Infinium HumanMethylation450 platform. With this function it is possible to obtain normalized data.

- beta_preprocessSWAN <- getBeta(preprocessSwan_results)
- beta_preprocessSWAN_I <- beta_preprocessSWAN[rownames(beta_preprocessSWAN) %in% dfI$IlmnID,]
- beta_preprocessSWAN_II <- beta_preprocessSWAN[rownames(beta_preprocessSWAN) %in% dfII$IlmnID,]
- mean_of_beta_preprocessSWAN_I <- apply(beta_preprocessSWAN_I,1,mean)
- mean_of_beta_preprocessSWAN_II <- apply(beta_preprocessSWAN_II,1,mean)
- d_mean_of_beta_preprocessSWAN_I <- density(mean_of_beta_preprocessSWAN_I,na.rm=T)
- d_mean_of_beta_preprocessSWAN_II <- density(mean_of_beta_preprocessSWAN_II,na.rm=T)
- sd_of_beta_preprocessSWAN_I <- apply(beta_preprocessSWAN_I,1,sd)
- sd_of_beta_preprocessSWAN_II <- apply(beta_preprocessSWAN_II,1,sd)
- d_sd_of_beta_preprocessSWAN_I <- density(sd_of_beta_preprocessSWAN_I,na.rm=T)
- d_sd_of_beta_preprocessSWAN_II <- density(sd_of_beta_preprocessSWAN_II,na.rm=T)

 # PLOT THE DISTRIBUTION

- par(mfrow=c(2,3))
- plot(d_mean_of_beta_I,col="blue",main="raw beta",xlim=c(0,1),ylim=c(0,6))
- lines(d_mean_of_beta_II,col="red")
- plot(d_sd_of_beta_I,col="blue",main="raw sd",xlim=c(0,0.6),ylim=c(0,90))
- lines(d_sd_of_beta_II,col="red")
- boxplot(beta,ylim=c(0,1))
- plot(d_mean_of_beta_preprocessSWAN_I,col="blue",main="preprocessSWAN beta",xlim=c(0,1),ylim=c(0,6))
- lines(d_mean_of_beta_preprocessSWAN_II,col="red")
- plot(d_sd_of_beta_preprocessSWAN_I,col="blue",main="preprocessSWAN sd",xlim=c(0,0.6),ylim=c(0,90))
- lines(d_sd_of_beta_preprocessSWAN_II,col="red")
- boxplot(beta_preprocessSWAN,ylim=c(0,1))



 # It is possible to observe that normalized data have β-values with mean distribution and sd distribution which are similar to those obtained from raw data.

8) Perform a PCA on the beta matrix.

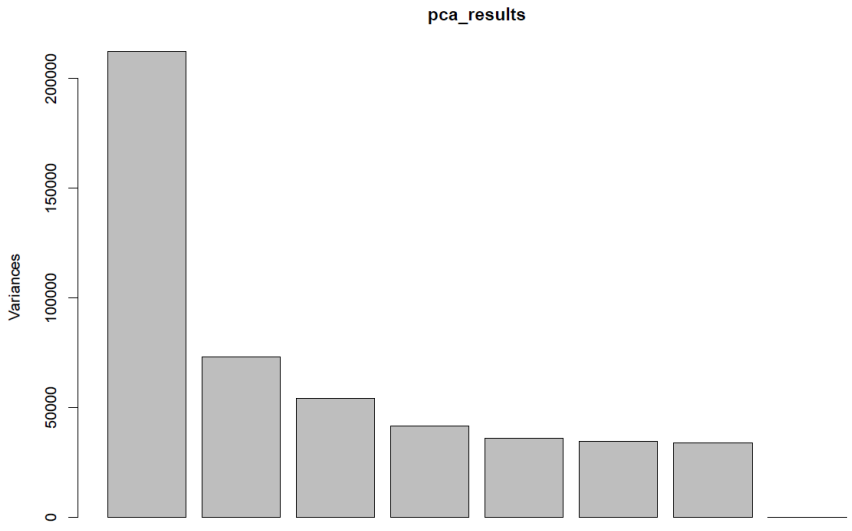
- pca_results <- prcomp(t(beta_preprocessSwan),scale=T)

 # The objective of PCA is to project points in multidimensional space onto a space of fewer dimension. It is possible to use this analysis to check if there are large differences in DNA methylation profiles in our samples.

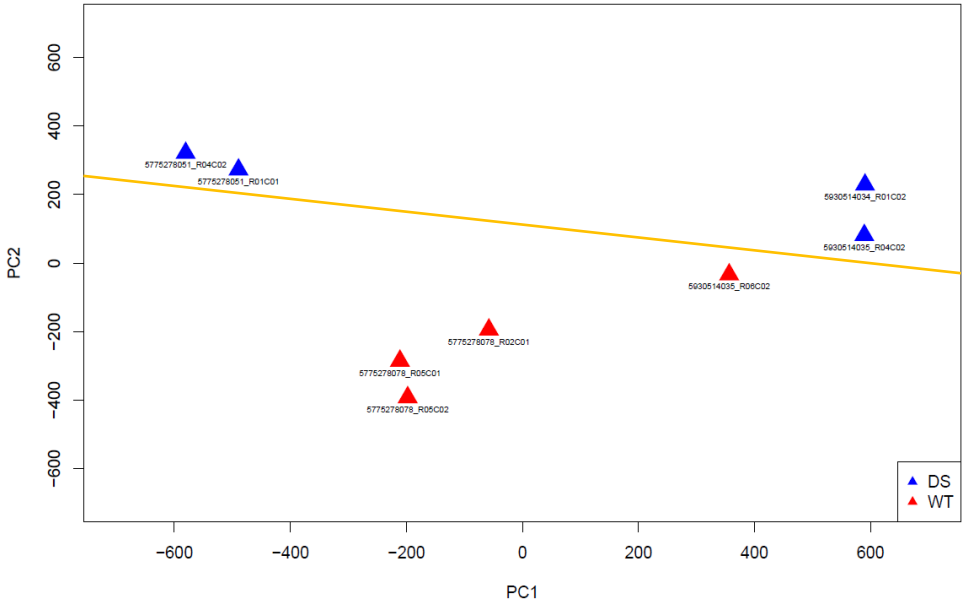
```
> print(summary(pca_results))
Importance of components:
                           PC1      PC2      PC3      PC4      PC5      PC6      PC7
Standard deviation     460.7490 270.1431 232.9555 203.94037 189.56530 186.09762 183.89622
Proportion of Variance   0.4373   0.1503   0.1118   0.08567   0.07401   0.07133   0.06965
Cumulative Proportion    0.4373   0.5876   0.6993   0.78500   0.85901   0.93035   1.00000
                           PC8
Standard deviation     6.053e-12
Proportion of Variance 0.000e+00
Cumulative Proportion  1.000e+00
```

- plot(pca_results)

 # Plot the variance accounted for each component



- pheno <- read.csv("Samplesheet_report_2020.csv",header=T, stringsAsFactors=T)
- palette(c("blue","red"))
- plot(pca_results$x[,1], pca_results$x[,2],cex=2,pch=17,col=pheno$Group,xlab="PC1",ylab="PC2",xlim=c(-700,700),ylim=c(-700,700))
- text(pca_results$x[,1], pca_results$x[,2],labels=rownames(pca_results$x),cex=0.5,pos=1)
- legend("bottomright",legend=levels(pheno$Group),col=c(1:nlevels(pheno$Group)),pch=17)



 # Dots in the plot are colored according to the group to which they belong. It is possible to identify a line (the yellow one in the graph above) which separate DS and WT dots.

9) Identify differentially methylated probes between group DS and group WT using the function Mann-Whitney test

- pheno <- read.csv("Samplesheet_report_2020.csv",header=T, stringsAsFactors=T)

```
> str(pheno)
'data.frame':   8 obs. of  6 variables:
 $ Sample_Name: int  1020 1036 3038 3042 3052 1016 1029 3029
 $ Group      : Factor w/ 2 levels "DS","WT": 1 1 2 2 2 1 1 2
 $ Age        : int  29 34 46 32 31 43 32 35
 $ Slide      : num  5.78e+09 5.78e+09 5.78e+09 5.78e+09 5.78e+09 ...
 $ Array      : Factor w/ 7 levels "R01C01","R01C02",..: 1 4 3 5 6 2 4 7
 $ Basename   : Factor w/ 8 levels "5775278051_R01C01",..: 1 2 3 4 5 6 7 8
```

- My_mannwhitney_function <- function(x) {
  wilcox <- wilcox.test(x~ pheno$Group)
  return(wilcox$p.value)
  }

 # Wilcox.test is a non-parametric test: if the data are paired it performs a Wilcoxon signed rank test, if the data are unpaired it performs a Mann-Whitney test. It is necessary to create a specific function because wilcox.test() function does not return a value but a list

- pValues_wilcox <- apply(beta_preprocessSwan,1, My_mannwhitney_function)
- final_wilcox <- data.frame(beta_preprocessSwan, pValues_wilcox)
- final_wilcox <- final_wilcox[order(final_wilcox$pValues_wilcox),]
- final_wilcox_0.05 <- final_wilcox[final_wilcox$pValues_wilcox<=0.05,]
- dim(final_wilcox_0.05)

#22381

10) Apply multiple test correction and set a significant threshold of 0.05. How many probes do you identify as differentially methylated considering nominal pValues? How many after Bonferroni correction? How many after BH correction?

- raw_pValues <- final_wilcox[,9]
- corrected_pValues_BH <- p.adjust(raw_pValues,"BH")
- corrected_pValues_Bonf <- p.adjust(raw_pValues,"bonferroni")

- final_wilcox_corrected <- data.frame(final_wilcox, corrected_pValues_BH, corrected_pValues_Bonf)
- dim(final_wilcox[final_wilcox$pValues_wilcox<=0.05,])

 # 22381 differentially methylated probes considering nominal p-Values

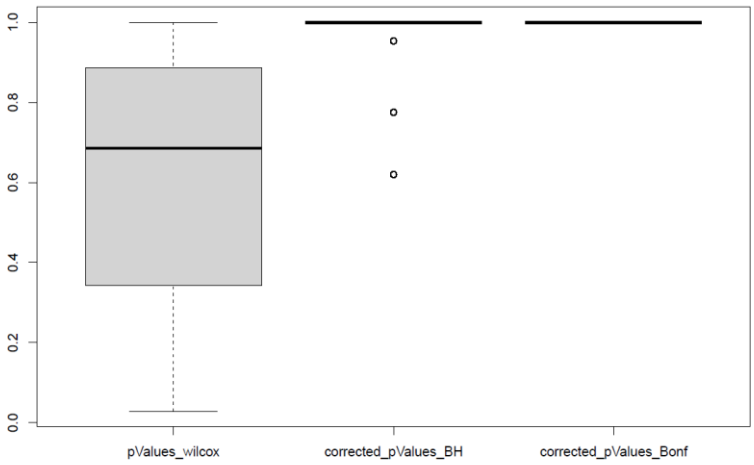
- dim(final_wilcox[final_wilcox$corrected_pValues_BH<=0.05,])

 # 0 differentially methylated probes considering p-Values after Bonferroni correction

- dim(final_wilcox[final_wilcox$corrected_pValues_Bonf<=0.05,])

 # 0 differentially methylated probes considering p-Values after BH correction

- boxplot(final_wilcox_corrected[,9:11])

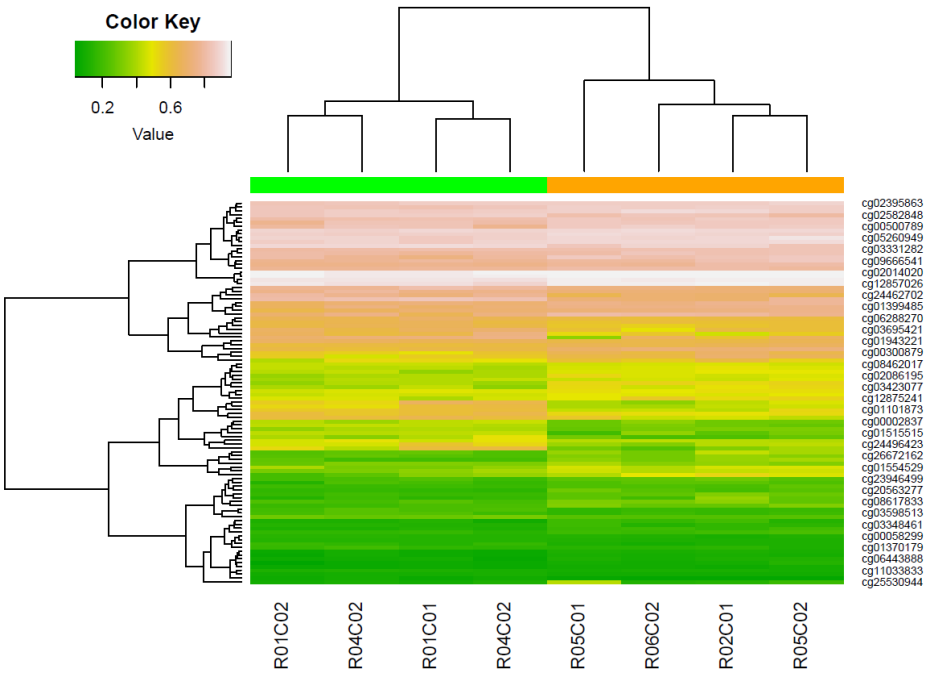


11) Produce an heatmap of the top 100 differentially mehtylated probes

- install.packages("gplots")
- library(gplots)
- input_heatmap=as.matrix(final_wilcox_corrected[1:100,1:8])

```
> pheno$Group
[1] DS DS WT WT WT DS DS WT
Levels: DS WT
```

- colorbar <- c("green","green","orange","orange","orange","green","green","orange")
- heatmap.2(input_heatmap,col=terrain.colors(100),Rowv=T,Colv=T,dendrogram="both",key=T,ColSideColors=colorbar,density.info="none",trace="none",scale="none",symm=F)



12) Produce a volcano plot and a Manhattan plot of the results of differential methylation analysis

 # VOLCANO PLOT

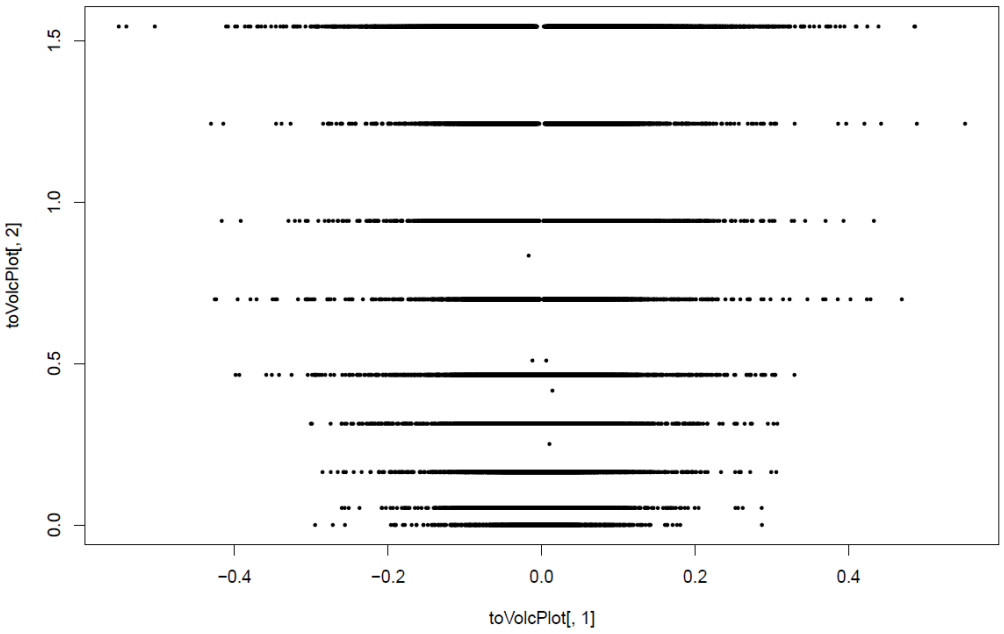
- beta <- final_wilcox_corrected[,1:8]
- beta_groupA <- beta[,pheno$Group=="DS"]
- mean_beta_groupA <- apply(beta_groupA,1,mean)
- beta_groupB <- beta[,pheno$Group=="WT"]
- mean_beta_groupB <- apply(beta_groupB,1,mean)

 # These steps are necessary to calculate the difference between the average of group DS and group WT values.

- delta <- mean_beta_groupB-mean_beta_groupA
- toVolcPlot <- data.frame(delta, -log10(final_wilcox_corrected$pValues_wilcox))

 # This dataframe has two columns, one contains the delta values and other contains the -10log of p-values.

- plot(toVolcPlot[,1], toVolcPlot[,2],pch=16,cex=0.5)



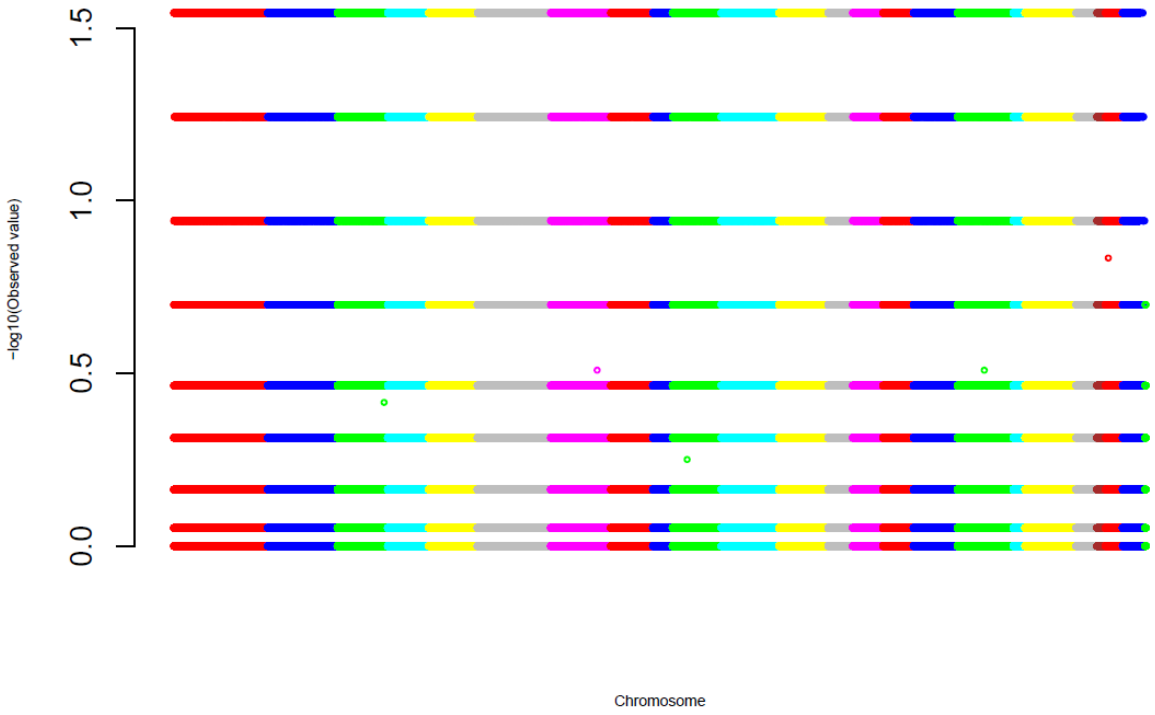
 # MANHATTAN PLOT

- install.packages("gap")
- library(gap)
- final_wilcox_corrected<- data.frame(rownames(final_wilcox_corrected),final_wilcox_corrected)

 # This step is necessary because in final_wilcox_corrected the CpG probes are stored in the row names, not in the columns.

```
> colnames(final_wilcox_corrected)
 [1] "rownames.final_wilcox_corrected." "X5775278051_R01C01"
 [3] "X5775278051_R04C02"               "X5775278078_R02C01"
 [5] "X5775278078_R05C01"               "X5775278078_R05C02"
 [7] "X5930514034_R01C02"               "X5930514035_R04C02"
 [9] "X5930514035_R06C02"               "pValues_wilcox"
[11] "corrected_pValues_BH"             "corrected_pValues_Bonf"
```

- colnames(final_wilcox_corrected)[1] <- "IlmnID"
- final_wilcox_corrected_annotated <- merge(final_wilcox_corrected, Illumina450Manifest_clean,by="IlmnID")
- input_Manhattan <- data.frame(final_wilcox_corrected_annotated$CHR, final_wilcox_corrected_annotated$MAPINFO, final_wilcox_corrected_annotated$pValues_wilcox)
- input_Manhattan$final_wilcox_corrected_annotated.CHR <- factor(input_Manhattan$final_wilcox_corrected_annotated.CHR,levels=c("1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","17","18","19","20","21","22","X","Y"))
- palette <-c ("red","blue","green","cyan","yellow","gray","magenta","red","blue","green","cyan","yellow","gray",
- "magenta","red","blue","green","cyan","yellow","gray","brown","red","blue","green")
- mhtplot(input_Manhattan,control=mht.control(colors=palette))
- axis(2,cex=0.5)



OPTIONAL: As DS is caused by the trisomy of chromosome 21, try also to plot the density of the methylation values of the probes mapping on chromosome 21. Do you see a very clear difference between the samples? How many differentially methylated probes do you find on chromosome 21?

- ID_CHR<-data.frame(Illumina450Manifest_clean$IlmnID, Illumina450Manifest_clean$CHR)
- cpg21=ID_CHR[ID_CHR$Illumina450Manifest_clean.CHR==21,]
- WT<-SampleSheet[SampleSheet$Group=="WT","Basename"]
- DS<-SampleSheet[SampleSheet$Group=="DS","Basename"]

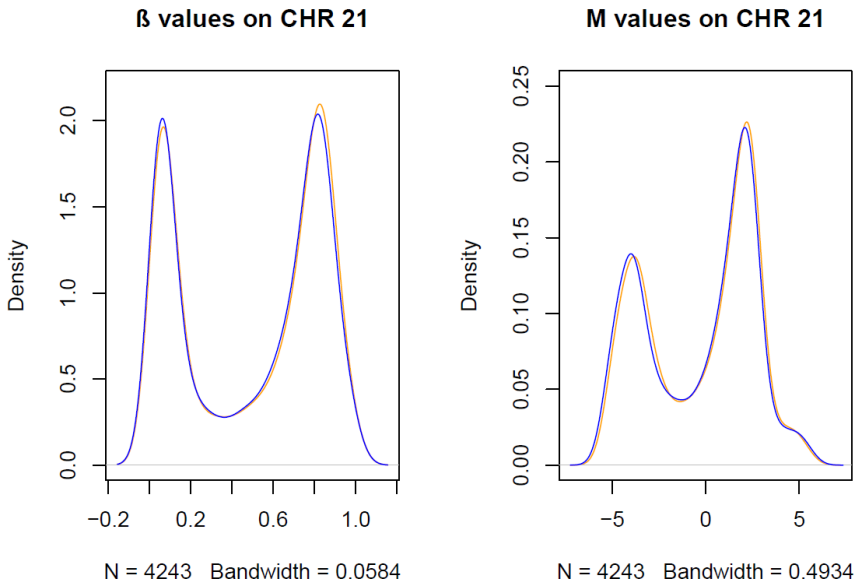
BETA VALUES

- beta <- getBeta(MSet.raw)
- beta_WT_21<- beta[rownames(beta)%in%cpg21$Illumina450Manifest_clean.IlmnID,colnames(beta)%in%WT]
- beta_DS_21<- beta[rownames(beta)%in%cpg21$Illumina450Manifest_clean.IlmnID,colnames(beta)%in%DS]
- mean_of_beta_WT_21<- apply(beta_WT_21,1,mean,na.rm=T)
- d_mean_of_beta_WT_21 <- density(mean_of_beta_WT_21)
- mean_of_beta_DS_21 <- apply(beta_DS_21,1,mean,na.rm=T)
- d_mean_of_beta_DS_21 <- density(mean_of_beta_DS_21)

M VALUES

- M <- getM(MSet.raw)
- M_WT_21<- M[rownames(M)%in%cpg21$Illumina450Manifest_clean.IlmnID,colnames(M)%in%WT]
- M_DS_21<- M[rownames(M)%in%cpg21$Illumina450Manifest_clean.IlmnID,colnames(M)%in%DS]
- mean_of_M_WT_21 <- apply(M_WT_21,1,mean,na.rm=T)
- d_mean_of_M_WT_21 <- density(mean_of_M_WT_21)
- mean_of_M_DS_21 <- apply(M_DS_21,1,mean,na.rm=T)
- d_mean_of_M_DS_21 <- density(mean_of_M_DS_21**)**

- par(mfrow=c(1,2))
- plot(d_mean_of_beta_WT_21,main="β values on CHR 21",col="orange",ylim=c(0,2.2))
- lines(d_mean_of_beta_DS_21, col="blue")
- plot(d_mean_of_M_WT_21,main="M values on CHR 21",col="orange",ylim=c(0,0.25))
- lines(d_mean_of_M_DS_21, col="blue")



 # Focusing on chromosome 21, there are no significant differences between the distributions.

- final_wilcox_0.05_21 <- final_wilcox[final_wilcox$pValues_wilcox<=0.05 & rownames(final_wilcox)%in%cpg21$Illumina450Manifest_clean.IlmnID,]
- dim(final_wilcox_0.05_21)

#301