# DNA Methylation Biomarker Discovery

**\*What is it and why is it necessary?**

DNA methylation is the process which stops transcription due to addition of methyl groups to the DNA chains. The addition can also act as biomarkers which further provide a brief idea on diseases especially cancers. Specific DNA methylation patterns can help to provide the onset or even the type of cancer along with its severity.

**GOAL**

Identify DNA methylation markers distinguishing healthy vs diseased samples.

**TOOLS**

R (Bioconductor Package), Python, Differential Analysis

**STEPS FOLLOWED:**

**1. Downloading Dataset**

A. The methylated dataset was downloaded from [GEO (Gene Expression Omnibus’s) official website.](https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE53051) For this project beta-values were downloaded, now beta-values are the numeric values recorded by probes in the flow cell when it finds methylation sites so, increase in the beta-values denotes an intensified methylation site.

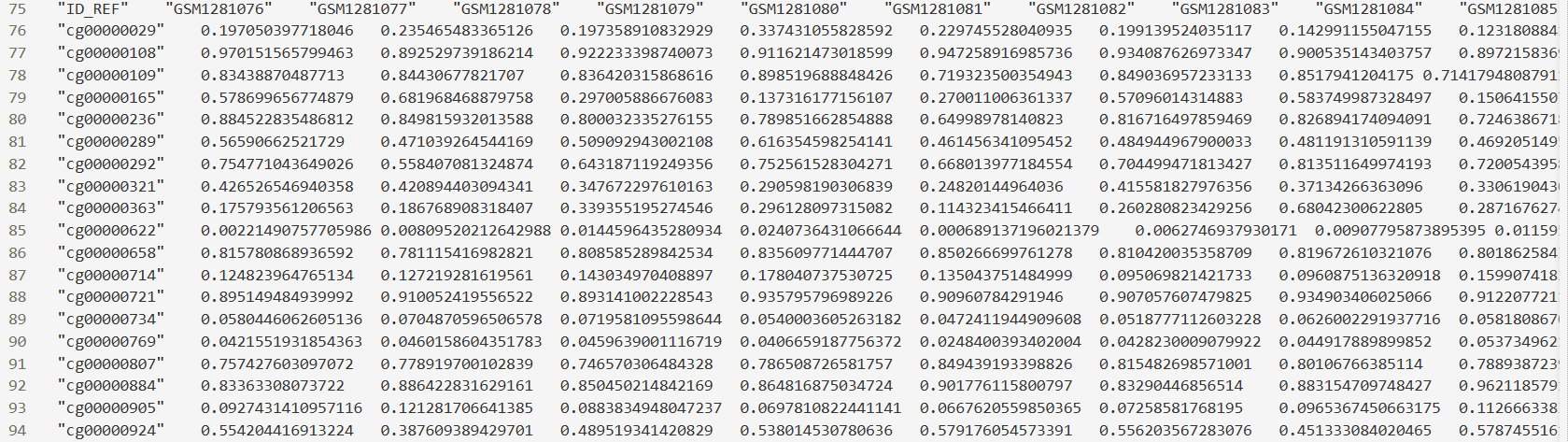
B. After navigating to website and scrolling down to the utmost end of the webpage, the hyperlink under the name of “[Series Matrix File(s)](https://ftp.ncbi.nlm.nih.gov/geo/series/GSE53nnn/GSE53051/matrix/)” contains the required data. After clicking on the hyperlink a new tab opens and the file titled “[GSE53051\_series\_matrix.txt.gz](https://ftp.ncbi.nlm.nih.gov/geo/series/GSE53nnn/GSE53051/matrix/GSE53051_series_matrix.txt.gz)” holds our target dataset which are the beta-values of methylation sites.

**Nature of the data: High-Dimensional**

**Size: 1.80 GBs**

**Unnecessary Rows from the Header: 74 (Required Data starts from the 75th Row)**

The image below depicts a small section of the dataset:



“ID\_REF”: Identifier for a CpG site (Cytosine-phosphorylated group-Guanosine) which are the sites where DNA methylation occurs.

GSM…. : SampleIDs having values between 0 to 1 indicating beta values.

**2. Installing Packages**

A. R-studio has been used which is ideal for data-rich projects as this one.

B. All the necessary packages have been installed by running the following commands in R’s console:

install.packages(“BiocManager”)

BiocManager::install(“GEOquery”)

BiocManager::install(“minfi”)

BiocManager::install(“ChAMP”)

**3. Loading the Data into R:**

This step helped to load the high dimensional data into R’s console for further processing:

The code lines to do so are as follows:

#Loading Methylation Data

matrix\_file = "C:/Users/mohak/Desktop/GSE53051\_series\_matrix.txt"

library(data.table) # Load the data.table package

library(GEOquery) # Load GEOquery for phenotype data

skip\_lines = 74 #Skips lines starting with "!"

cat(paste0("\nLoading ", matrix\_file))

#TRY CATCH

beta\_values\_dt = tryCatch({

fread(

input = matrix\_file,

skip = skip\_lines,

header = TRUE,

data.table = FALSE, # Output as a standard data.frame

check.names = FALSE,

verbose = TRUE, # Keep verbose for detailed messages

nThread = 1

)

},

error = function(e) {

message(paste0("Error during fread"))

NULL

})

# Check if beta\_values\_dt was successfully created

if (is.null(beta\_values\_dt))

{

stop("fread failed to load the data.")

} else {

cat("\nSuccessfully loaded data using fread!\n")

}

# Renaming the first column to "ID\_REF" which are row names

if ("ID\_REF" %in% colnames(beta\_values\_dt))

{

rownames(beta\_values\_dt) = beta\_values\_dt$ID\_REF

beta\_values\_dt$ID\_REF = NULL # Remove the column as it's now row names

cat("Set 'ID\_REF' as row names and removed the column.\n")

}else{

first\_col\_name = colnames(beta\_values\_dt)[1]

if (!is.null(first\_col\_name) && startsWith(first\_col\_name, "cg")) #CpG IDs start with "cg"

{

cat(paste0("First column is '", first\_col\_name, "'. Setting as row names.\n"))

rownames(beta\_values\_dt) = beta\_values\_dt[[first\_col\_name]]

beta\_values\_dt[[first\_col\_name]] = NULL

} else {

warning("Column 'ID\_REF' not found.")

}

}

#Analyzing the Data Frame

cat("\nDimensions of the beta value matrix:\n")

print(dim(beta\_values\_dt))

cat("\nFirst 5 rows and first 5 columns of the beta value matrix:\n")

print(head(beta\_values\_dt[, 1:min(5, ncol (beta\_values\_dt))]))

# Identify a sample ID from the header for summary

sample\_id\_for\_summary = colnames(beta\_values\_dt)[1] # Take the first sample column

if (!is.null(sample\_id\_for\_summary))

{

cat(paste0("\nSummary of beta values for the first sample (", sample\_id\_for\_summary, "):\n"))

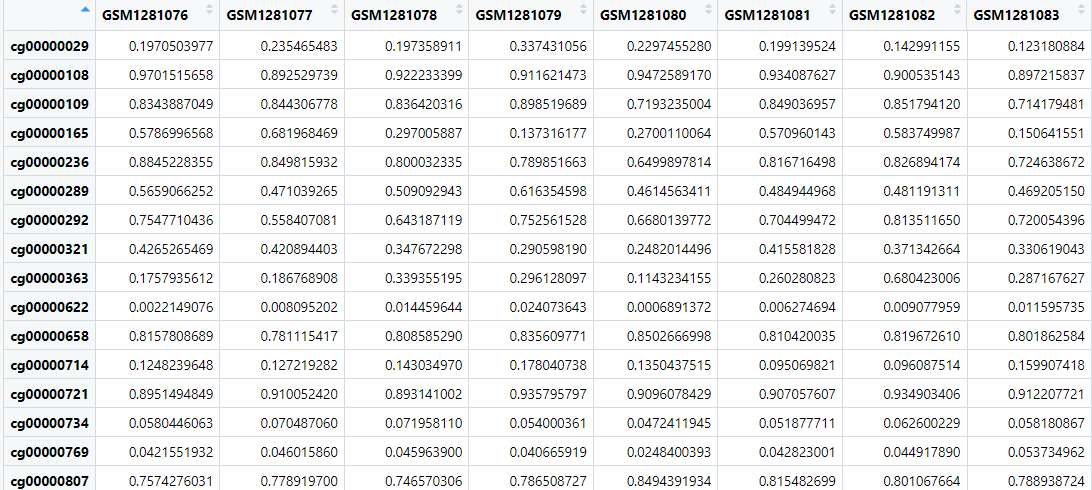
print(summary(beta\_values\_dt[, sample\_id\_for\_summary]))

} else {

warning("Could not identify a sample column for summary.")

}

The Following Image depicts the snippet of the loaded dataset “beta\_values\_dt”:



**Dimensions: Rows** 485512

**Columns** 220

**4. Creating a Phenotypic Dataframe**

This particular dataframe titled “pheno\_data” helps to align the SampleIDs with their metadata meaning all the sample related information such as its tissue source, characteristics (normal or cancerous), the DNA extraction protocol and the labelling protocol has been obtained for each individual sample. All of this will be used further in the project.

The following code lines help to create the dataframe:

gse\_local = getGEO(filename = "C:/Users/mohak/Desktop/GSE53051\_series\_matrix.txt", GSEMatrix = TRUE, getGPL = FALSE)

if (class(gse\_local) == "list") {

gse\_local = gse\_local[[1]] # Select the first (and likely primary) GSMList if it's a list

}

**# The phenotype data is stored in the pData slot of the ExpressionSet object**

pheno\_data = pData(phenoData(gse\_local))

cat("\nDimensions of the phenotype data:\n")

print(dim(pheno\_data))

cat("\nFirst few rows and selected columns of phenotype data:\n")

print(head(pheno\_data[, c("title", "source\_name\_ch1", "characteristics\_ch1.1", "characteristics\_ch1", "characteristics\_ch1.2")]))

cat("\nAvailable columns in pheno\_data:\n")

print(colnames(pheno\_data))

**#Aligning Phenotypic Data columns with original dataset**

sample\_names\_beta = colnames(beta\_values\_dt)

sample\_names\_pheno = rownames(pheno\_data)

common\_samples = intersect(sample\_names\_beta, sample\_names\_pheno)

if (length(common\_samples) == 0) {

stop("No common samples found between beta values and phenotype data.")

}

if (length(common\_samples) < length(sample\_names\_beta)) {

warning(paste0("Number of common samples (", length(common\_samples), ") is less than total beta samples (", length(sample\_names\_beta), ")."))

}

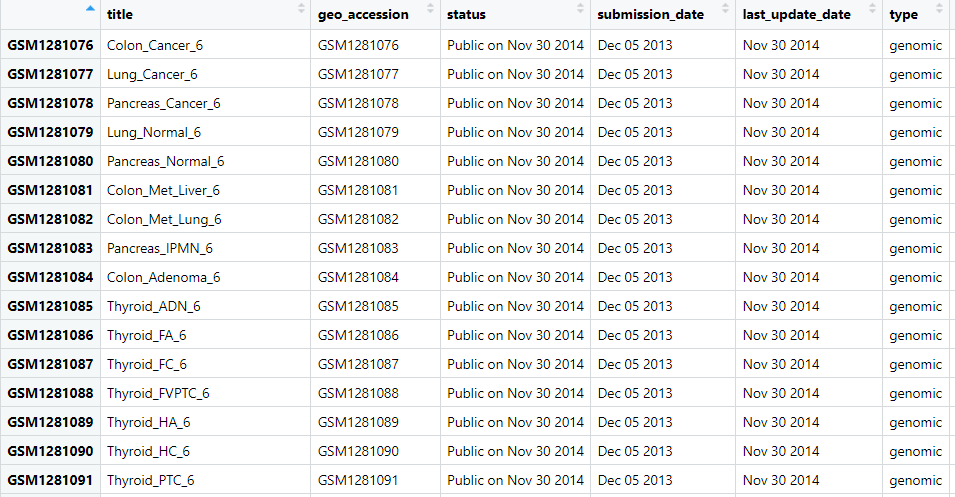
**# Reordering original dataset columns and pheno\_data rows to match**

beta\_values\_dt = beta\_values\_dt[, common\_samples]

pheno\_data = pheno\_data[common\_samples, ]

print("\nSuccessfully aligned ")

The image below depicts a small section of the entire dataframe:



**Dimensions:** Rows 220

Columns 47

**5. Preparing pheno\_data (Phenotypic Dataframe for differential analysis)**

In this the phenotypic dataframe is prepared, cleaned and centred around two binary values which give the overall state of the disease which are “Normal” and “Cancer”. The “Adenoma”, “Metastasis” have not been considered because the overall sample size of the differential analysis reduces, which would give inaccurate results.

The following code lines help to do so:

**# 1. Standardizing Disease State**

pheno\_data$Condition = tolower(pheno\_data$`characteristics\_ch1.1`)

pheno\_data$Condition[grepl("normal", pheno\_data$Condition)] = "normal"

pheno\_data$Condition[grepl("cancer|tumor|adenocarcinoma", pheno\_data$Condition)] = "cancer" # Added common cancer terms

pheno\_data$Condition = factor(pheno\_data$Condition, levels = c("normal", "cancer")) # Only normal/cancer as main levels

**# 2. Standardizing Source Tissue**

pheno\_data$Tissue = tolower(gsub(" tissue|\\s+sample|\\s+tissue", "", pheno\_data$source\_name\_ch1, ignore.case = TRUE))

pheno\_data$Tissue = factor(pheno\_data$Tissue)

#Filtering and Realigning

filtered\_sample\_ids = rownames(pheno\_data)[pheno\_data$Condition %in% c("normal", "cancer") &

!is.na(pheno\_data$Condition) & !is.na(pheno\_data$Tissue)]

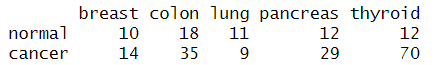
pheno\_data\_filtered = pheno\_data[filtered\_sample\_ids, ]

beta\_values\_dt\_filtered = beta\_values\_dt[, filtered\_sample\_ids]

**#Filtered dataframe**

cat("\nSample Distribution (Normal vs. Cancer) by Tissue:\n")

print(table(pheno\_data\_filtered$Condition, pheno\_data\_filtered$Tissue))

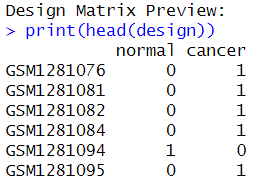
****

The image above depicts the number of methylation data samples available for each of the cancer type. Out of them the most balanced sample length for both cancerous and normal ones is for colon so, further methylation analysis on colon cancer’s beta-values has been done.

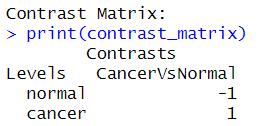
**6. Differential Analysis**

For this we have used a Bioconductor package entitled “limma”, this package needs to know which binary group is the reference one (keep in mind that we have “Normal” and “Cancer” as our binary groups) so, we will set “Normal” group as our reference. This is done in order to let the ‘logFC’ (log fold change) represent the methylation level in the ‘cancer’ group (which is what we need) in the form of (Methylation[Cancer] – Methylation[Normal]). A positive logFC means hypermethylation and a negative logFC means hypomethylation in ‘cancer’ group.

A design matrix is important to let limma understand the relationships between our samples. So, in short design matrix creates sampled groups of our colon cancer. Meaning our SampleIDs would have the number of samples which are cancerous or normal in a tabular format due to the design matrix.



The contrast matrix is defined to make specific comparisons between these groups meaning it highlights the difference in methylation between normal and cancer groups.



**Note: Differential Analysis initially has been done on Colon cancer as it had largest samples.**

Number of significant DMPs (adj. P < 0.05): 152910

Number of significant DMPs with |delta beta| > 0.1: 93354

**DMPs: DNA Methylation Probes**

The code lines help to perform differential analysis on the beta-values of colon cancer:

1. Install Packages:

BiocManager::install(“limma”,force=TRUE)

BiocManager::install(“IlluminaHumanMethylation450kanno.ilmn12.hg19",force=TRUE)

library(limma)

library(minfi) # For getting probe annotations

library(IlluminaHumanMethylation450kanno.ilmn12.hg19) # Annotation data

cat("\nDifferential Methylation Analysis\n")

#Subset data for Colon tissue

chosen\_tissue = "colon"

colon\_pheno = pheno\_data\_filtered[pheno\_data\_filtered$Tissue == chosen\_tissue, ]

colon\_beta = beta\_values\_dt\_filtered[, rownames(colon\_pheno)]

cat(paste0("\nAnalyzing ", chosen\_tissue, " tissue:\n"))

print(table(colon\_pheno$Condition))

#Normal as reference

colon\_pheno$Condition = relevel(colon\_pheno$Condition, ref = "normal")

#Defining the Design Matrix

design = model.matrix(~0 + Condition, data = colon\_pheno)

colnames(design) = gsub("Condition", "", colnames(design))

cat("\nDesign Matrix Preview:\n")

print(head(design))

cat("\nDimension of Design Matrix:", dim(design), "\n")

#Contrast matrix for comparing Cancer with Normal

contrast\_matrix = makeContrasts(CancerVsNormal = cancer - normal, levels = design)

cat("\nContrast Matrix:\n")

print(contrast\_matrix)

#Limma

#Fitting the linear model

cat("\nFitting linear model\n")

fit = lmFit(colon\_beta, design)

#Applying contrasts

cat("Applying contrasts\n")

fit2 = contrasts.fit(fit, contrast\_matrix)

#Smoothening

cat("Performing empirical Bayes\n")

fit2 = eBayes(fit2)

print("\nLimma Completed\n")

# 5. Extract results and add annotations

topDMPs = topTable(fit2, coef = "CancerVsNormal", number = Inf, adjust.method = "BH") # Inf to get all results

# Adding annotations to the results

cat("\nAdding probe annotations...\n")

data(IlluminaHumanMethylation450kanno.ilmn12.hg19) #Gets annotation Data for Illumina 450k

anno = getAnnotation(IlluminaHumanMethylation450kanno.ilmn12.hg19)

#Merging

anno\_subset = anno[, c("chr", "pos", "UCSC\_RefGene\_Name", "Relation\_to\_Island", "Regulatory\_Feature\_Group")]

topDMPs\_annotated = merge(topDMPs, anno\_subset, by = "row.names", all.x = TRUE)

colnames(topDMPs\_annotated)[1] = "Probe\_ID" # Rename the first column back to Probe\_ID

cat("\nFirst few rows of Top Differentially Methylated Probes (DMPs):\n")

print(head(topDMPs\_annotated))

cat("\nSummary of results:\n")

#Adjusted p Values to find out significance

significant\_DMPs = subset(topDMPs\_annotated, adj.P.Val < 0.05)

cat(paste0("Number of significant DMPs (adj. P < 0.05): ", nrow(significant\_DMPs), "\n"))

# Biologically Significant |delta beta| > 0.1

significant\_and\_large\_effect\_DMPs = subset(significant\_DMPs, abs(logFC) > 0.1) # delta beta > 0.1 (10% difference)

cat(paste0("Number of significant DMPs with |delta beta| > 0.1: ", nrow(significant\_and\_large\_effect\_DMPs), "\n"))

#Beta Values for significant DMPs

cat("\nSummary of delta beta values for significant DMPs:\n")

print(summary(significant\_DMPs$logFC))

#Save the files:

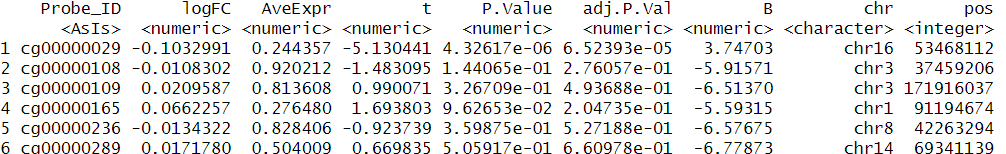
library(openxlsx)

output\_file1 = "C:/Users/mohak/Desktop/Colon Cancer(Binary) All probes.xlsx"

output\_file2 = "C:/Users/mohak/Desktop/Colon Cancer(Binary) significant beta.xlsx"

write.xlsx(topDMPs\_annotated, file = output\_file1, rownames=FALSE)

write.xlsx(significant\_and\_large\_effect\_DMPs, file= output\_file2, rownames=FALSE)



The image above shows the top-ranked probes by significance:

Probe\_ID: Identifier for each CpG site now, this is the site in which Cytosine is immediately followed by Guanine with a phosphorylated backbone and these are the sites at which DNA methylation occurs .

logFC: This is the difference in average beta values between the cancer and normal binary categories

The negative logFC in “1” denotes Hypomethylation which denotes lower methylation in cancer than normal, keep in mind we had selected normal as the reference to find out the average beta values in the cancer category. CG00000029 with a -0.103 logFC denotes a -10.3% less methylation in cancer.

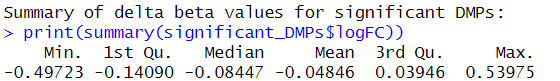
Adjusted P-value using Benjamini-Hochberg P-value adjustment which helps to identify which results are truly significant.

****

This indicates a very large number of statistically significant differentially methylated probes. So, 152910 are methylation differences at specific CpG sites which are possible to exist in real-life scenario and biologically relevant.



Same for the difference in beta values. 93354 are methylation differences at CpG sites are biologically possible. The delta beta is the actual magnitude or size of the methylation difference between the binary groups. Here the threshold is >0.1 because of the scale. Beta values range between 0 and 1 which represent a biological significance so, a significance threshold as less as 0.05 would not help to understand the truly significant data which defining a threshold of 0.1 does and helps to mark a biomarker as useful.



**7. Visualization:**

The differential data has been visualized in the form of volcano plot and heatmap in order to get a better understanding on the nature of the samples.

1. **Volcano Plot**

BiocManager::install(“EnhancedVolcano”)

install.packages(“dplyr”)

library(openxlsx)

library(dplyr) # For data manipulation like filter

file\_path = "C:/Users/mohak/Desktop/Colon Cancer(Binary) All probes.xlsx"

topDMPs\_annotated = read.xlsx(file\_path)

# Ensuring logFC and adj.P.Val are numeric, as read.xlsx might sometimes read them as character

topDMPs\_annotated$logFC = as.numeric(topDMPs\_annotated$logFC)

topDMPs\_annotated$adj.P.Val = as.numeric(topDMPs\_annotated$adj.P.Val)

# Re-creating significant\_and\_large\_effect\_DMPs for use in the plot caption

significant\_and\_large\_effect\_DMPs = topDMPs\_annotated %>%

filter(adj.P.Val < 0.05, abs(logFC) > 0.1)

cat("topDMPs\_annotated file loaded successfully. Dimensions: ")

print(dim(topDMPs\_annotated))

#Volcano Plot

cat("\nGenerating Volcano Plot...\n")

library(EnhancedVolcano)

library(ggplot2)

volcano\_plot = EnhancedVolcano(

toptable = topDMPs\_annotated,

lab = topDMPs\_annotated$Probe\_ID, # Label with Probe IDs

x = 'logFC',

y = 'adj.P.Val',

pCutoff = 0.05,

FCcutoff = 0.1,

title = 'Colon Cancer vs. Normal Colon: Differential Methylation',

subtitle = 'Identified by limma-eBayes',

# Using the re-created significant\_and\_large\_effect\_DMPs for the caption counts

caption = paste0('Total DMPs = ', nrow(topDMPs\_annotated),

'\nSignificant (adj. P < 0.05) = ', nrow(subset(topDMPs\_annotated, adj.P.Val < 0.05)),

'\nSignificant & Delta Beta > 0.1 = ', nrow(significant\_and\_large\_effect\_DMPs)),

pointSize = 1.0,

labSize = 3.0,

legendPosition = 'right',

col = c('plum4', 'maroon', 'mediumpurple2', 'coral2'),

colAlpha = 0.5,

drawConnectors = TRUE,

widthConnectors = 0.5,

colConnectors = 'grey30'

)

print("\nPrinting Volcano Plot to RStudio Plots pane\n")

print(volcano\_plot) # This line makes the plot appear!

tryCatch({

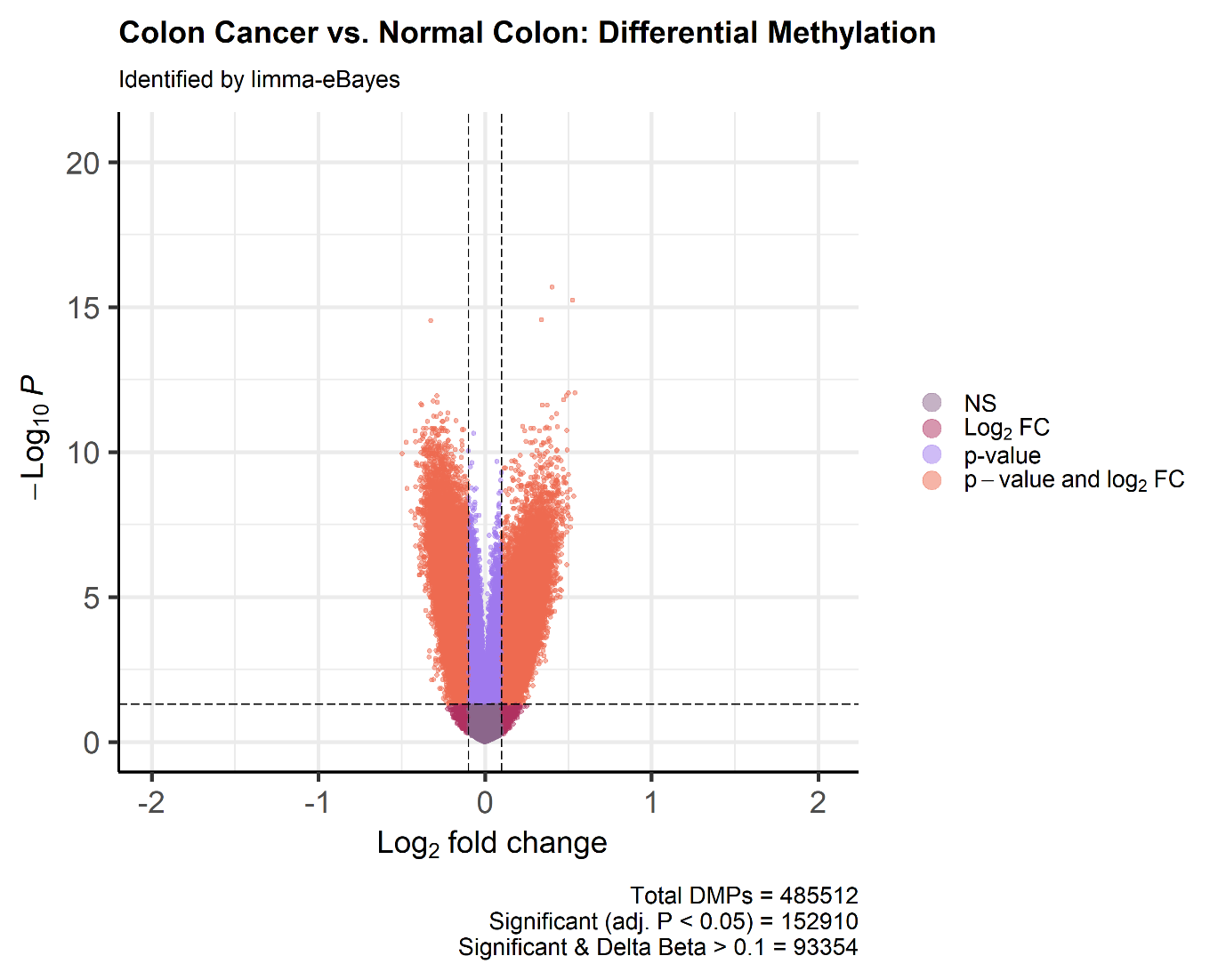
ggsave("C:/Users/mohak/Desktop/Colon\_Volcano\_All\_probes.png", plot = volcano\_plot, width = 10, height = 8, dpi = 300)

}, error = function(e) {

message("Error saving the plot")

})

cat("\nFinished Volcano Plot operations.\n")



* The x-axis represents the delta beta values in a Log2 fold change which further help to classify hypermethylated and hypomethylated samples. In this, the central two vertical lines (denote biological significance) represent |logFC| > 0.1 (Right to 0) and |logFC| < 0.1 (Left to 0) and the samples which lie on the right side are hypermethylated (cancerous) which are contradictory to those which lie on the left side (normal).
* The Y-axis represents Log10P wherein the horizontal dotted line denotes the threshold of adj.P.Val < 0.05 which is ~1.301 as Log10(0.05) ~ 1.301 so, realistically the samples which lie above this point are biologically significant (which have adj.P.Val < 0.05 example adj.P.Val = 0.04, Log10(0.04) ~1.39)
* Higher values on Y-axis increases the statistical significance.
* Plum Points: Not statistically Significant
* Coral Points: Both Biologically and Statistically Significant
* Medium Purple Points: Statistically significant but extremely low probabilities to exist.
* Maroon: Not Statistically significant, False Positives.

**B. Heatmap**

library(openxlsx)

library(dplyr)

library(pheatmap)

library(limma)

library(minfi)

# Loading topDMPs\_annotated

file\_path\_dmp = "C:/Users/mohak/Desktop/Colon Cancer(Binary) All probes.xlsx"

topDMPs\_annotated = read.xlsx(file\_path\_dmp)

topDMPs\_annotated$logFC = as.numeric(topDMPs\_annotated$logFC)

topDMPs\_annotated$adj.P.Val = as.numeric(topDMPs\_annotated$adj.P.Val)

significant\_and\_large\_effect\_DMPs = topDMPs\_annotated %>%

filter(adj.P.Val < 0.05, abs(logFC) > 0.1)

# Selecting top 1001 DNA methylation Probes

# Sorting by adjusted p-value and taking the top 1001

if(nrow(significant\_and\_large\_effect\_DMPs) >= 1001) {

top\_N\_dmps = significant\_and\_large\_effect\_DMPs %>%

arrange(adj.P.Val) %>%

head(1001)

} else {

top\_N\_dmps = significant\_and\_large\_effect\_DMPs

warning("Less than 1,001 significant DMPs found. Plotting all ", nrow(top\_N\_dmps), " DMPs.")

}

sig\_dmps = top\_N\_dmps$Probe\_ID

# Loading Beta Values Data

file\_path\_beta = "C:/Users/mohak/Desktop/GSE53051\_series\_matrix.txt"

beta\_matrix = read.delim(file\_path\_beta, skip = 74, row.names = 1, stringsAsFactors = FALSE)

colnames(beta\_matrix) = gsub("\"","", colnames(beta\_matrix))

cat("Beta matrix loaded. Dimensions: ")

print(dim(beta\_matrix))

# Phenotype Data

file\_path\_pheno = "C:/Users/mohak/Desktop/GSE processed Phenotype data.xlsx"

pheno = read.xlsx(file\_path\_pheno)

colon\_pheno = pheno %>%

select(

`geo\_accession`,

`source\_name\_ch1`,

`state:ch1`

) %>%

filter(`source\_name\_ch1` == "colon") # Filter using original name

# Setting 'normal' as reference level for Condition factor

colon\_pheno$`state:ch1` = as.factor(colon\_pheno$`state:ch1`)

colon\_pheno$`state:ch1` = relevel(colon\_pheno$`state:ch1`, ref = "normal")

cat("Processed phenotype data (colon\_pheno) head: \n")

print(head(colon\_pheno))

# Heatmap Data Preparation

# Using 'geo\_accession' for sample names

beta\_samples = beta\_matrix[, colon\_pheno$`geo\_accession`]

# sig\_dmps is now defined by the top N selection

heatmap\_data = beta\_samples[sig\_dmps, ] # Ensure sig\_dmps comes from the top N selection

cat("Heatmap data dimensions (probes x samples): ")

print(dim(heatmap\_data))

# Preparing Sample Annotations for pheatmap

# Using 'state:ch1' for Condition

annotation\_col = data.frame(Condition = colon\_pheno$`state:ch1`)

# Using 'geo\_accession' for rownames

rownames(annotation\_col) = colon\_pheno$`geo\_accession`

annotation\_colors = list(

Condition = c(normal = "maroon", cancer = "lightslateblue")

)

cat("Annotation column head: \n")

print(head(annotation\_col))

heatmap\_colors = colorRampPalette(c("maroon", "lavender", "lightslateblue"))(100)

#Heatmap

tryCatch(

{

pheatmap(

mat = heatmap\_data,

color = heatmap\_colors,

annotation\_col = annotation\_col,

annotation\_colors = annotation\_colors,

cluster\_rows = TRUE,

cluster\_cols = TRUE,

show\_rownames = FALSE,

show\_colnames = TRUE,

scale = "row",

main = "Heatmap of Methylation Analysis (Top 1,001 DMPs)", # Edited title

filename = "C:/Users/mohak/Desktop/Colon\_Heatmap\_Top1001.png", # Edited filename

width = 10,

height = 12

)

cat("\nHeatmap generated and saved as Colon\_Heatmap\_Top1001.png\n")

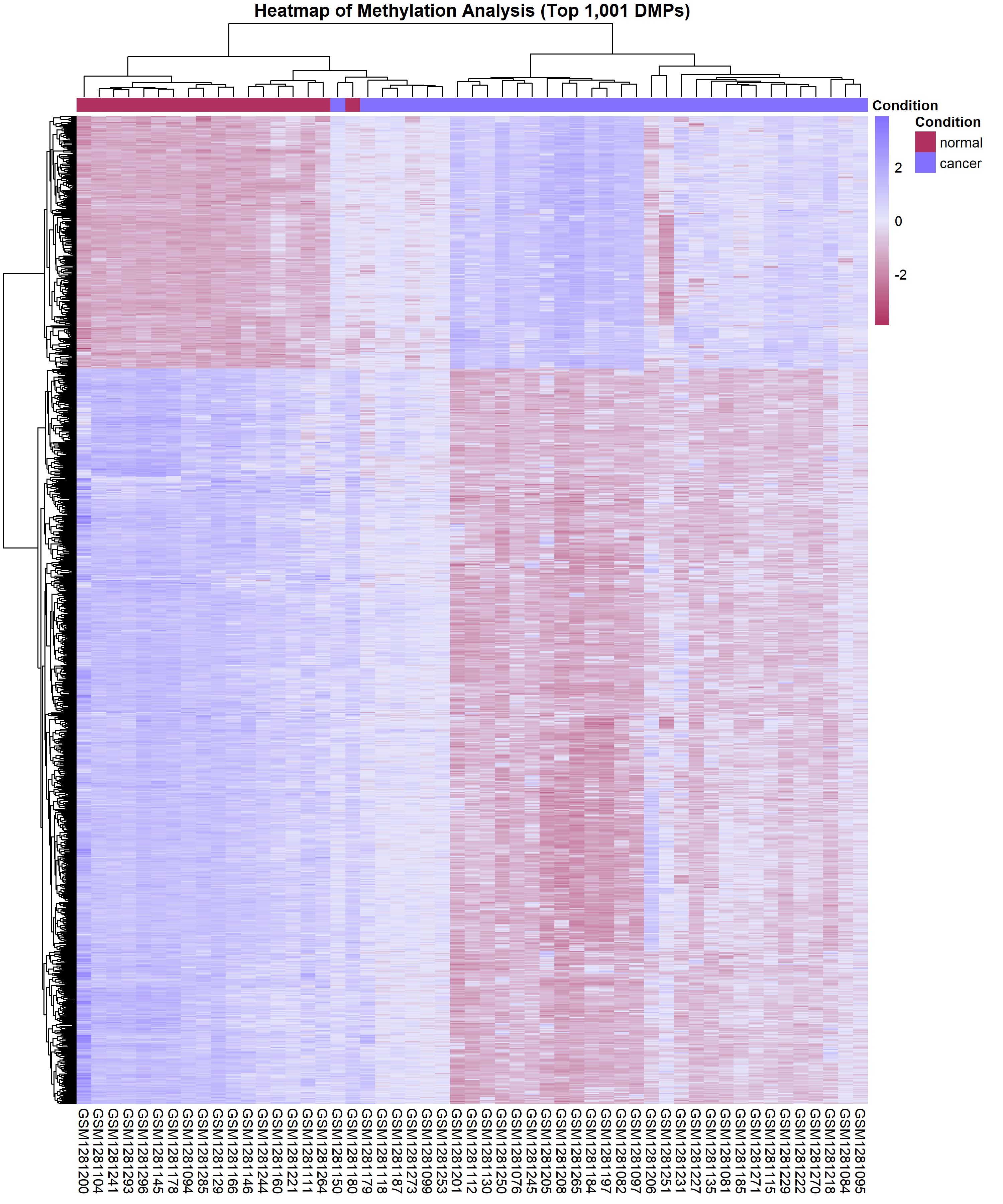
},

error = function(e) {

message("Error generating the heatmap: ", e$message)

}

)



* Each row is represented by an individual differentially methylated probe from the 1001. These are CpG sites which are both statistically and biologically significant.
* The vertical column represents an individual biological sample taken from the patient’s colon tissue. So, GSM are patient samples and each row of this sample represents individual CpG site methylation visual.
* Normal marker appears maroon in colour and Cancerous appear Lightslateblue. Lavender samples denotes a region between both of them meaning the sample is neither hypomethylated (normal) nor hypermethylated (cancer) but has a higher chance to be hypermethylated as CpG sites in cancerous tissues are generally hypermethylated.

**8. Extracting Significant CpG biomarkers**

In this we have created a file named “Colon Cancer(Binary) significant beta.xlsx” which holds the significant probes where methylation levels are statistically and significantly classify normal and cancer groups.

The process helps in disease association and understanding a high dimensional data statistically.

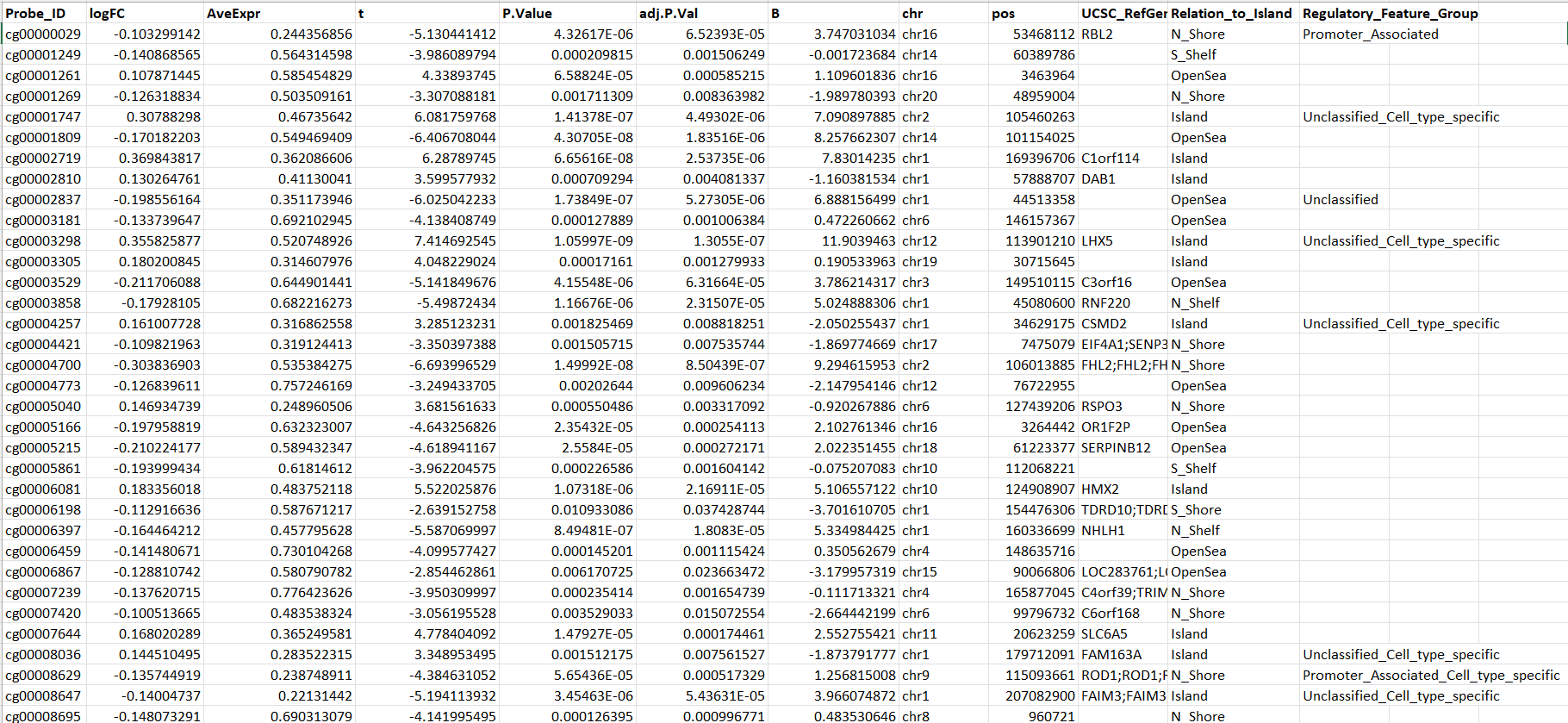
Already done while saving the respective output files in

“Save the Files” step by running the following code:

output\_file2 = "C:/Users/mohak/Desktop/Colon Cancer(Binary) significant beta.xlsx"

write.xlsx(significant\_and\_large\_effect\_DMPs, file= output\_file2, rownames=FALSE)

**A small section of the Dataframe:**



**Dimensions:** Rows 93355

Columns 12

**9. Documenting the findings with gene annotations**

We have analyzed each and every methylation value but a complete dataset is needed to complete the project, the aim of this dataset is to provide the methylation analysis of the most significant data along with their genetic locations and characteristics.

The following code lines help to do so:

library(dplyr)

library(IlluminaHumanMethylation450kanno.ilmn12.hg19)

library(openxlsx)

# Annotation Data

# Using getAnnotation() from minfi on the annotation object

# This extracts a data frame with probe IDs as rownames and annotation as columns

annotation\_450k = getAnnotation(IlluminaHumanMethylation450kanno.ilmn12.hg19)

cat("Annotation data Dimensions: ")

print(dim(annotation\_450k))

# Selecting relevant annotation columns (e.g., gene symbol, chromosome, relation to CpG island)

relevant\_annotations = annotation\_450k %>%

as.data.frame() %>% # Converting to data frame to use dplyr functions on it

dplyr::select(

`UCSC\_RefGene\_Name`, # Gene symbol

`UCSC\_RefGene\_Group`,# Relation to gene

`Relation\_to\_Island`, # Relation to CpG island

`chr`, # Chromosome

`pos` # Genomic position

)

cat("Head of relevant\_annotations: \n")

print(head(relevant\_annotations))

# Merging the significant DMPs with the annotation data

# Joining by Probe\_ID (from DMPs) and rownames (from annotations)

# Converting rownames of relevant\_annotations to a column for merging

relevant\_annotations$Probe\_ID = rownames(relevant\_annotations)

file\_path\_significant\_DMPs = "C:/Users/mohak/Desktop/Colon Cancer(Binary) significant beta.xlsx"

significant\_and\_large\_effect\_DMPs = read.xlsx(file\_path\_significant\_DMPs)

cat("Loaded significant DMPs dimensions: ")

print(dim(significant\_and\_large\_effect\_DMPs))

# Performing the left\_join

significant\_DMPs\_with\_genes = significant\_and\_large\_effect\_DMPs %>%

dplyr::left\_join(relevant\_annotations, by = "Probe\_ID")

cat("Dimensions of annotated DMPs: ")

print(dim(significant\_DMPs\_with\_genes))

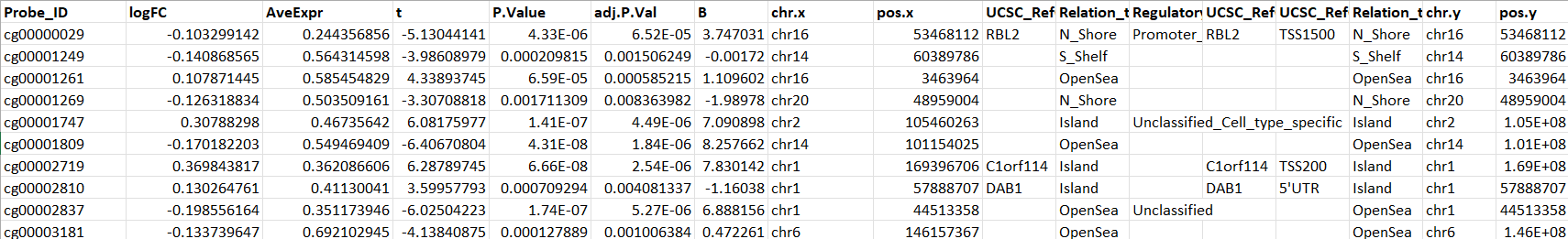
# New Dataframe

write.csv(significant\_DMPs\_with\_genes,

file = "C:/Users/mohak/Desktop/Gene Annotations of Significant biomarkers.csv",

row.names = FALSE

The image below showcases the new and complete dataset:



**Dimensions: Rows** 93355

**Columns** 17

1. Probe\_ID: Each row in this column is a potential CpG biomarker with its LogFC value.
2. LogFC: This quantifies the magnitude and direction of the methylation difference between our normal and cancer sample. The positive values denote Hypermethylation and negative values denote hypomethylation.
3. AveExpr: This is the average beta value across all the samples for a particular CpG site.
4. T: Statistical test which indicates the strength of evidence.
5. P.Value: The unadjusted p-value for differential methylation analysis.
6. Adj.P.Val: Value less than 0.05 denotes statistical significance.
7. B: Statistic test which provides differential expression by incorporating p-value and fold change.
8. Chr.x: Denotes the chromosome being considered
9. Pos.x: Denotes the base pair position of CpG site on the genome. (Applicable for chromosome x).
10. UCSC\_RefGene\_Name.x: This is the gene symbol that each CpG site is associated with meaning many CpG sites are located within promoters or bodies and knowing the name of the gene in which they are located would help to understand the disease pattern or the biological function caused due to the methylation change. (Applicable for chromosome x)
11. Relation\_to\_Island.x: Denotes the relationship of the CpG site to CpG islands which are high density CpG sites often found in gene promoters and their values individually represent:  
    A. Island: within a CpG island

B. N\_Shore (North\_Shore), S\_Shore (South\_Shore): Regions within 2kb of a CpG island. Located 2 Kilobases from the island boundary.

C. N\_Shelf (North\_Shelf), S\_Shelf (South\_Shelf): Regions 2-4kb from a CpG island. CpG sites located 2 to 4 kilobases away from the CpG island.

D. OpeanSea: Isolated CpG sites outside of islands, shores or shelves. Not located within the CpG island, Shelve, Shore.

The use of such parameters is to denote that methylation patterns and their functional consequences can vary significantly depending on the CpG’s island, shelve, shore or opensea nature. This in short tells us where the CpG site is located relative to the CpG island. CpG islands are important because they are usually unmethylated in healthy cells and abnormal methylation can help to identify the cancer as the methylation suppresses the Tumour Suppressing genes leading to uncontrolled expression of the tumour growth gene.

**\*CONCLUSION:**Appropriate data harvesting, processing and differential methods have helped to analyze and process the high dimensional data and achieve the ultimate objective of the project.