WGBS Analysis Pipeline

Mark E. Pepin, MS, PhD

21 January, 2021

Contents

Author: Mark E. Pepin, MS, PhD Biomedical Engineering | MD-PhD Trainee
Contact: pepinme@gmail.com
Institution: University of Alabama at Birmingham
Location: 542 Biomedical Research Building 2, Birmingham, AL 35294

#Genome Assembly and Alignment

The first task is to align the bisulfite reduced and sequenced reads to a genome assembly. To accomplish this, the genome assembly was generated based on Gencode annotation (gencode.v28.annotation.gtf) and sequence (GRCh38.p12.genome.fa). For whole-genome bisulfite sequencing via the Bismark (v0.20.0) aligner and genome preparation, a CT- and GA-converted assemblies are created.

##Genome Assembly

##Adapter and Read Quality Trimming

Once the genome assembly was created, adapter sequences were trimmed and sequencing quality assessed via trim_galore and FastQC, respectively.

```
trim_galore -o $INPUT_DIR/fastq_trimmed/ --paired --rrbs --non_directional --length 20
--fastqc #$INPUT_DIR/fastq/${VAR}_R1_001.fastq.gz $INPUT_DIR/fastq/${VAR}_R2_001.fastq.gz.
```

##Read Alignment

We then aligned all .fastq files to the genome assemblies using the following command:

```
bwameth.py --threads 8 --reference $GENOME_DIR/GRCh38.p12.genome.fa $INPUT_DIR/fastq_trimmed/${VAR}_R1_0
$INPUT_DIR/fastq_trimmed/${VAR}_R2_001_val_2.fq.gz > $RESULTS_DIR/RRBS_bwa/${VAR}.sam
```

Once aligned, we converted to .bam output.

```
#Convert .sam to .bam samtools view -S -b $RESULTS_DIR/RRBS_bwa/${VAR}.sam > $RESULTS_DIR/RRBS_bwa/${VAR
#Sort using samtools samtools sort $RESULTS_DIR/RRBS_bwa/${VAR}.bam -o $RESULTS_DIR/RRBS_bwa/${VAR}.sort
#create an index   samtools index $RESULTS_DIR/RRBS_bwa/${VAR}.sorted.bam   #MethylDackel
MethylDackel extract $GENOME_DIR/GRCh38.p12.genome.fa $RESULTS_DIR/RRBS_bwa/${VAR}.sorted.bam
-o $RESULTS_DIR/RRBS_bwa/${VAR}.counted --methylKit
```

Once finished, the CpG methylation was extracted as both bedgraph file (for UCSC genome browser) and bed file, which was then used to identify differentially-methylated cytosines (DMCs) and differentially-methylated regions (DMRs).

The "*.counted" files that resulted from this process were then read into R () and combined into a single “object” for differential methylation analysis

#Differential Methylation Analysis

##Combining sample methylation

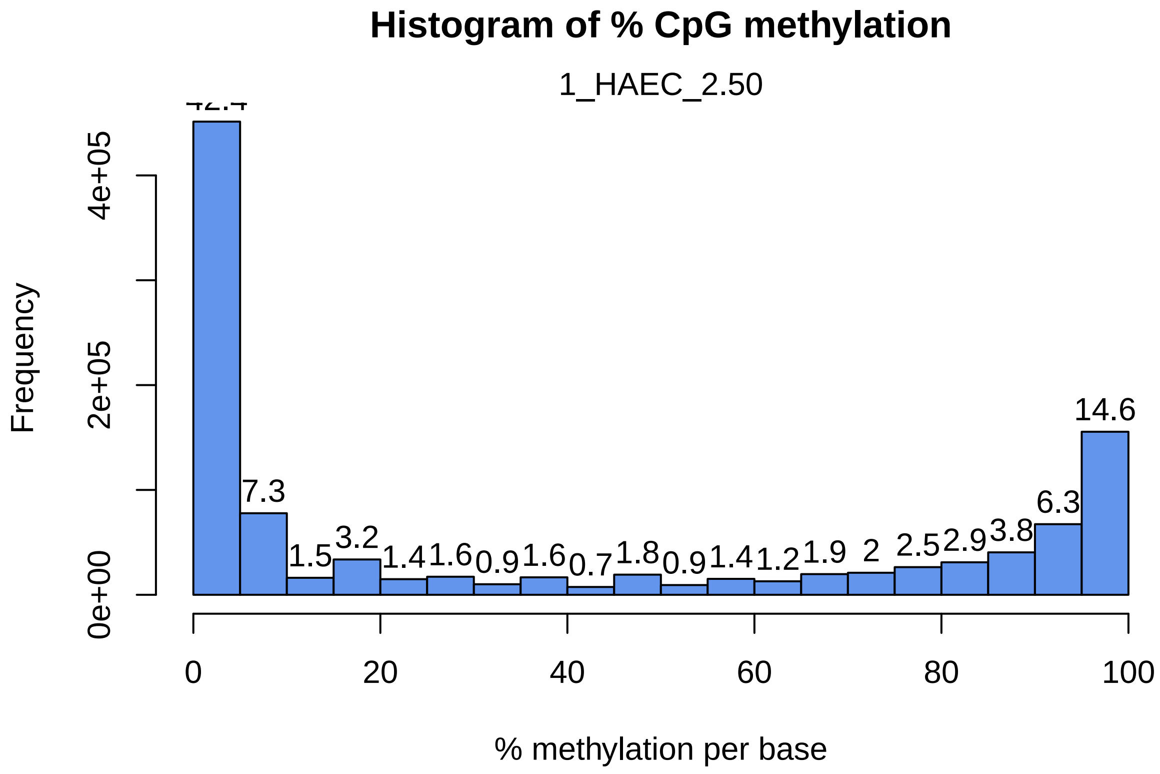
```
#Conditions to be used in differential methylation analysis (FILL OUT)
library(openxlsx)
library(dplyr)
TREATMENT=c("CON","HIGH")
CELL=c("HAEC")

ANALYSIS="HAEC_DM.v.CON"
### "2" is Pre-LVAD, "3" is Post-LVAD, "1" is CON
library(methylKit)
file.list <- list.files(path = paste0("../1_Input/Methyl/2_bwameth.out/", CELL), pattern = "*.counted_Cp
#Generate Column names (remove the extra nonsense from the path names)
colnames <- gsub( "*.counted_CpG.bedGraph", "", file.list)
colnames <- gsub( paste0("[.][.]/1_Input/Methyl/2_bwameth.out/", CELL, "/"), "", colnames)
sample_id<-as.list(colnames)
#Import the Index file
Index.raw<-read.xlsx("../1_Input/Index/Index_Napoli.xlsx", sheet = CELL)
Index.raw$Treatment<-factor(Index.raw$Group, levels = c("CON", "LOW", "HIGH"))

## Sort the index according to the .bed file ordering (as imported). This is important for correct anno
Index.raw$Treatment<-as.integer(Index.raw$Treatment)
Index.raw<-Index.raw %>% arrange(Sample.ID)

## Filter according to analysis parameters (CELL and TREATMENT)
Index.subset<-subset(Index.raw, Group %in% TREATMENT)
sample_id<-as.list(colnames)
##Create a methlRawlistDB
file.list<-as.list(file.list)
myobj<-methRead(file.list, sample.id = sample_id, assembly = "hg38", treatment = Index.raw$Treatment, p
##Example of smaple statistics (can spot check these)
getMethylationStats(myobj[[2]], plot = F, both.strands = F)
```

```
## methylation statistics per base
## summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   5.556  31.377  75.000 100.000
## percentiles:
##         0%        10%        20%        30%        40%        50%        60%
##   0.000000   0.000000   0.000000   0.000000   0.000000   5.555556  19.047619
##        70%        80%        90%        95%        99%      99.5%      99.9%
##  57.692308  85.714286 100.000000 100.000000 100.000000 100.000000 100.000000
##       100%
## 100.000000
```

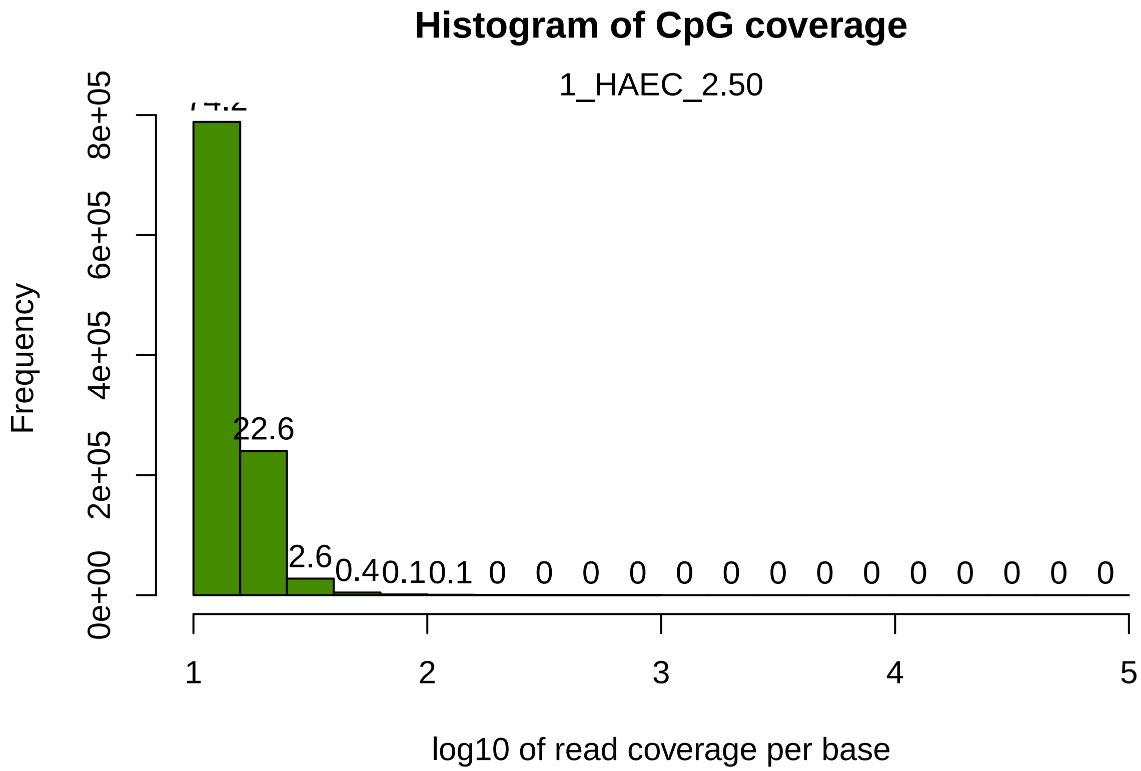
```
#Subset the methylRawList to include only the sample_id's for the desired analysis
myobj_filtered<-reorganize(myobj, sample.ids = Index.subset$Sample.ID, Index.subset$Treatment)
```

Once the samples have been compiled, it is valuable to perform some basic visualizations and statistics to determine whether quality filtering is necessary. The distribution of methylation change is plotted as a histogram (typically bimodal at the extremes), as well as a distribution of the read coverage per based, again plotted as a histogram. For the latter plot, it is important to determine whether PCR duplication biases the read coverage. If so, a secondary peak would emerge on the right-most portion of the histogram. In the current analysis, coverage distribution exhibits a one-tailed distribution, suggesting that the "deduplication" step in the alignment effectively eliminated the PCR amplification bias in coverage.

```r
library(graphics)
getMethylationStats(myobj_filtered[[2]], plot = T, both.strands = F)
```



```r
getCoverageStats(myobj_filtered[[2]], plot = T, both.strands = F)
```



```
#Save these files in an output folder
ifelse(!dir.exists(file.path("../2_Output/", ANALYSIS)), dir.create(file.path("../2_Output/", ANALYSIS)
```

```
## [1] FALSE
```

```
pdf(file=paste0("../2_Output/", ANALYSIS, "/", ANALYSIS, "_Methylation.Stats.pdf"))
getMethylationStats(myobj_filtered[[2]], plot = T, both.strands = F)
dev.off()
```

```
## pdf
##   2
```

```
pdf(file=paste0("../2_Output/", ANALYSIS, "/", ANALYSIS, "_Coverage.Stats.pdf"))
getCoverageStats(myobj_filtered[[2]], plot = T, both.strands = F)
dev.off()
```

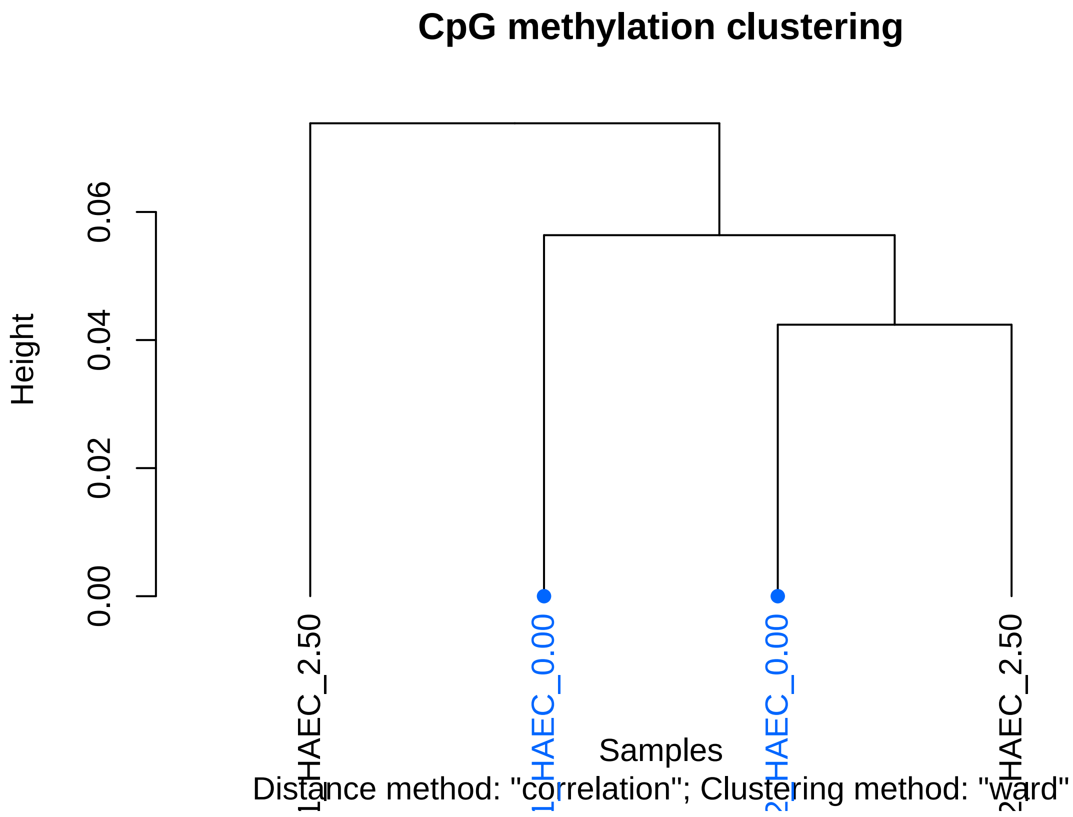
```
## pdf
##   2
```

Although most important in the context of correcting PCR-bias (duplication), filtering samples based on coverage also reduces false discovery based on low-coverage genomic regions. If PCR bias exists, an artificially high coverage would exist. Low coverage is also a concern due to low statistical power associated with low-coverage regions. Below, we discard bases with coverage below 10X, but also discard bases with coverage > 99.9th percentile.

```
#remove exceedingly high-coverage (risk of PCR bias) or low-coverage DMPs (low statistical power)
filtered.myobj <- filterByCoverage(myobj_filtered, lo.count = 5, lo.perc = NULL, hi.count = NULL, hi.pe
```

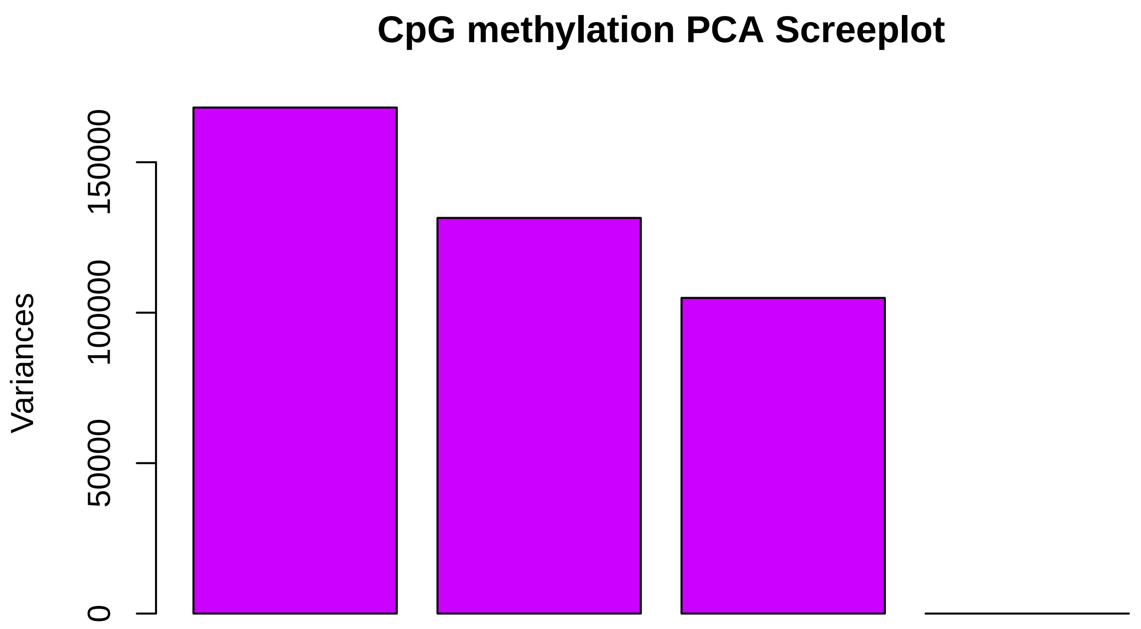
##Visualizing Methylation

```
#destrand and unite the sample data
meth<-unite(filtered.myobj, destrand = FALSE) #When calculating DMRs, it is not helpful to "destrand"
clusterSamples(meth, dist = "correlation", method = "ward", plot = TRUE)
```

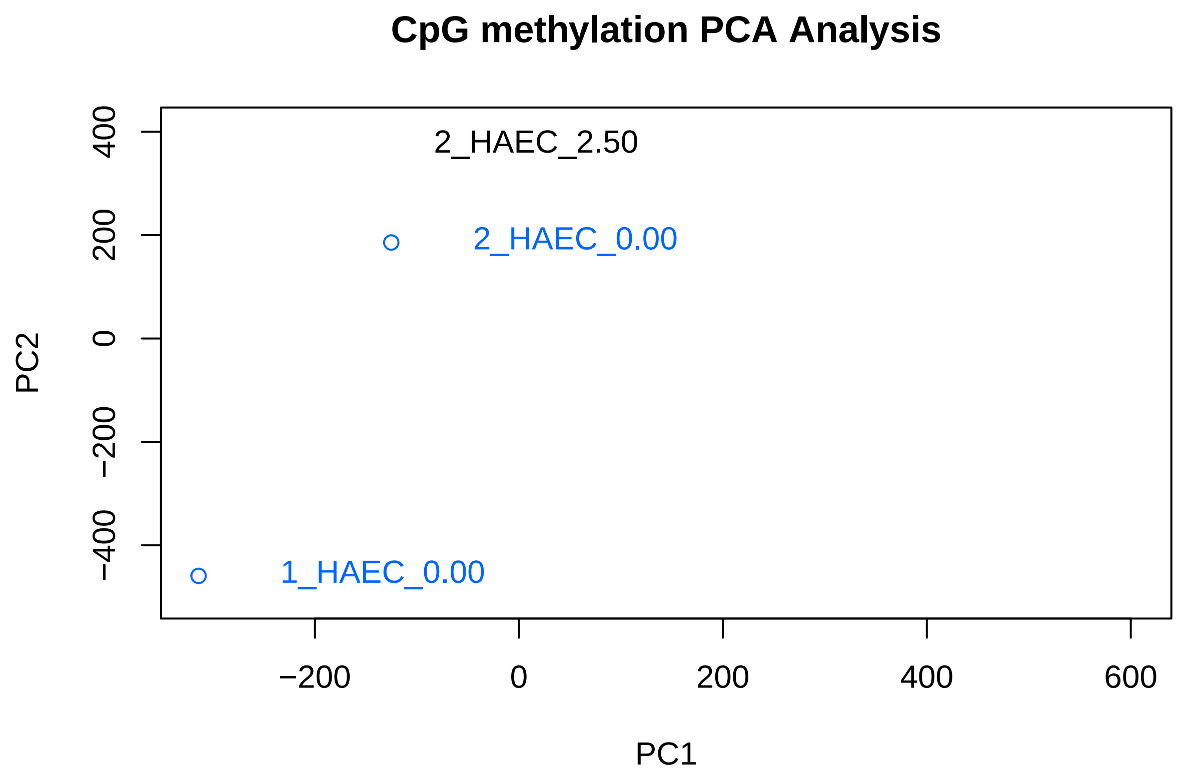


```
##
## Call:
## hclust(d = d, method = HCLUST.METHODS[hclust.method])
##
## Cluster method   : ward.D
## Distance         : pearson
## Number of objects: 4
```

```
PCASamples(meth, screeplot = TRUE)
```



```
PCASamples(meth)
```



```
#Create a folder in which to generate all documents/tables for this analysyis
ifelse(!dir.exists(file.path("../2_Output/", ANALYSIS)), dir.create(file.path("../2_Output/", ANALYSIS)
```

```
## [1] FALSE
```

```
#Create dendrogram and PCA plots
pdf(file=paste0("../2_Output/", ANALYSIS, "/", ANALYSIS, "_Clustering.pdf"))
clusterSamples(meth, dist = "correlation", method = "ward", plot = TRUE)
```

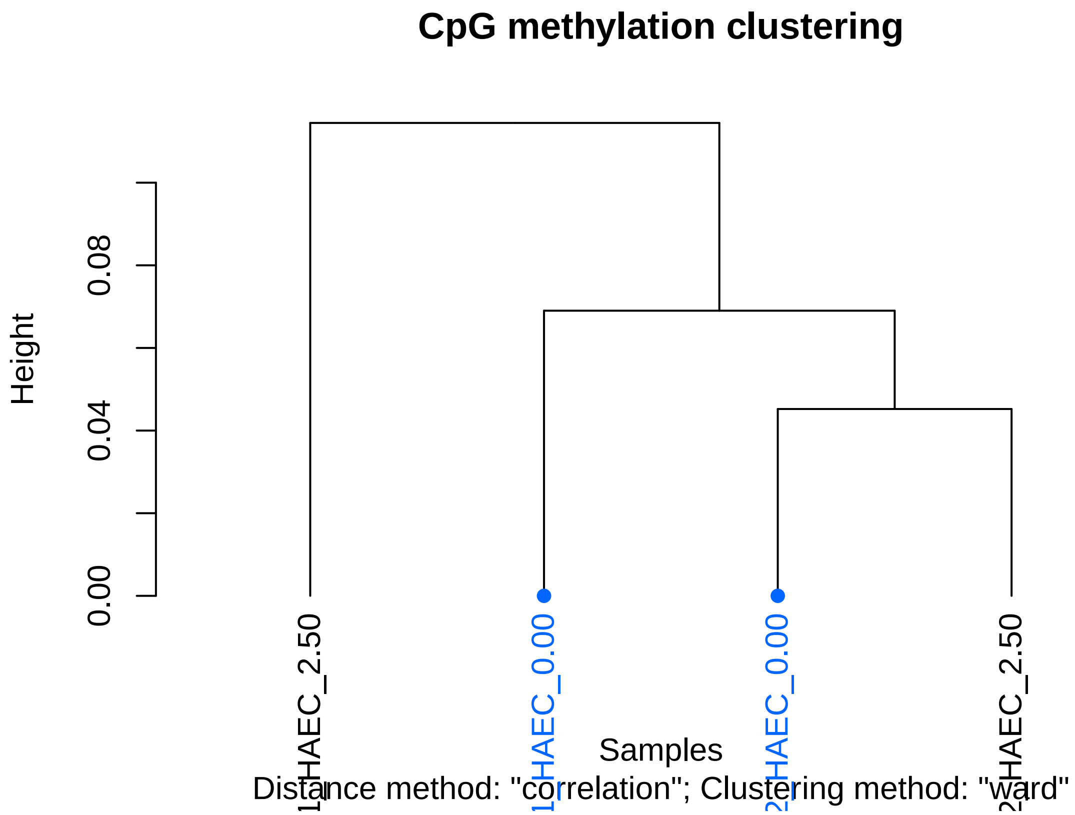
```
##
## Call:
## hclust(d = d, method = HCLUST.METHODS[hclust.method])
##
## Cluster method   : ward.D
## Distance         : pearson
## Number of objects: 4
```

```
PCASamples(meth, screeplot = TRUE)
PCASamples(meth)
dev.off()
```

```
## pdf
##   2
```

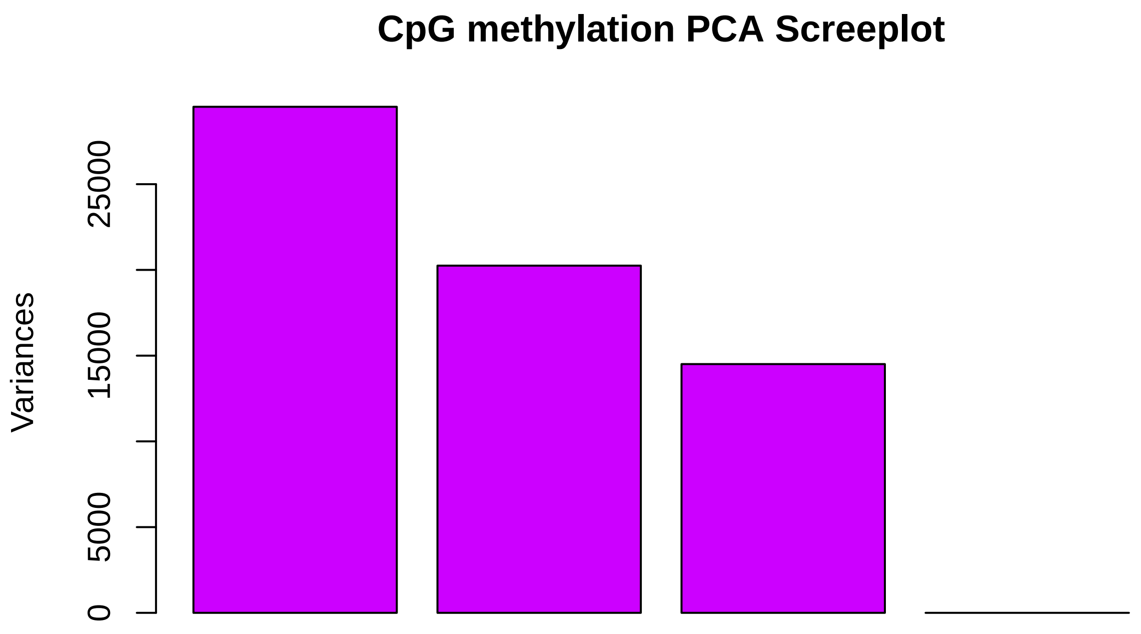
##Tiling Methylation Windows

```
library(dplyr)
###################################
#Differential Methylation of Tiles
###################################
tiles = tileMethylCounts(myobj_filtered, win.size = 500, step.size = 500)
meth_tile<-unite(tiles, destrand = FALSE) #When calculating DMRs, it is not helpful to "destrand"
myDiff_tiles=calculateDiffMeth(meth_tile, test = "Chisq", adjust = "SLIM", mc.cores = 7)
myDiff_tile.md<-as(myDiff_tiles,"methylDiff")
myDiff_tiles.filtered<-getData(myDiff_tile.md)

#Check clustering of samples by DMR correlation
clusterSamples(meth_tile, dist = "correlation", method = "ward", plot = TRUE)
```

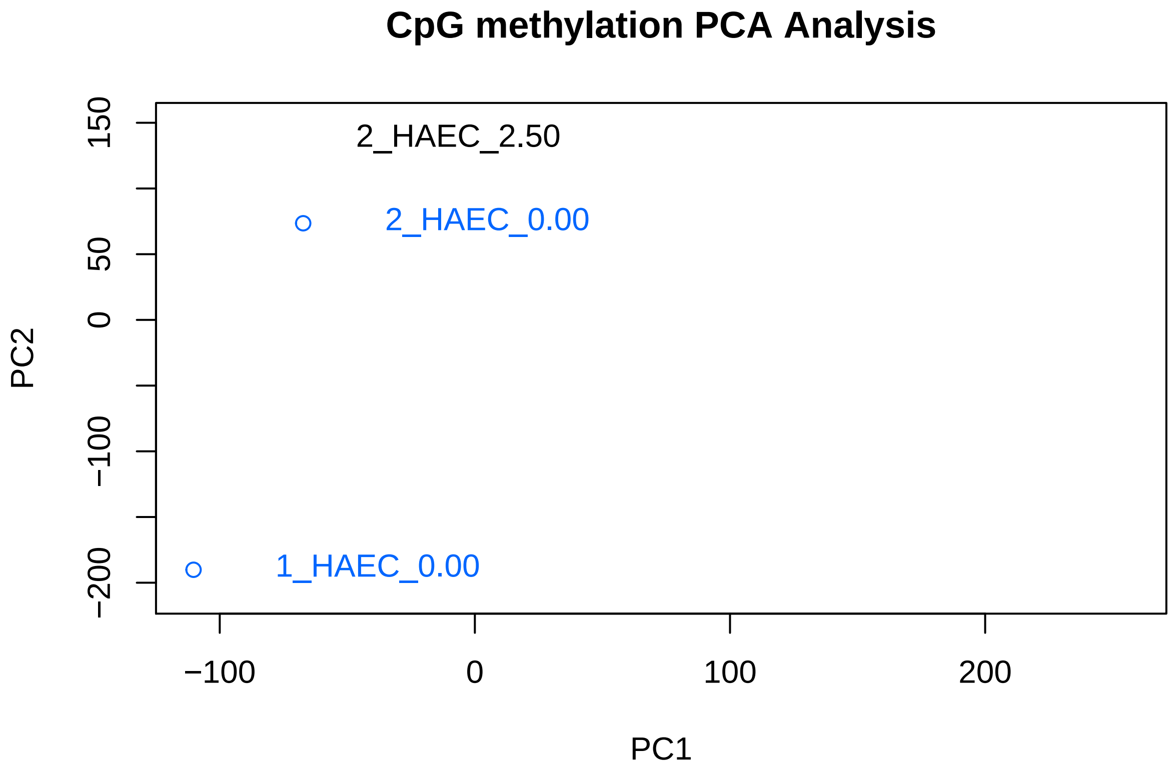


```
##
## Call:
## hclust(d = d, method = HCLUST.METHODS[hclust.method])
##
## Cluster method   : ward.D
## Distance         : pearson
## Number of objects: 4
```

```
PCASamples(meth_tile, screeplot = TRUE)
```



```
PCASamples(meth_tile)
```



```
#Create a folder in which to generate all documents/tables for this analysyis
ifelse(!dir.exists(file.path("../2_Output/", ANALYSIS, "DMR/")), dir.create(file.path("../2_Output/", AN
```

```
## [1] FALSE
```

```
#Create dendrogram and PCA plots
pdf(file=paste0("../2_Output/", ANALYSIS, "/DMR/", ANALYSIS, "_Clustering.pdf"))
clusterSamples(meth_tile, dist = "correlation", method = "ward", plot = TRUE)
```

```
##
## Call:
## hclust(d = d, method = HCLUST.METHODS[hclust.method])
##
## Cluster method   : ward.D
## Distance         : pearson
## Number of objects: 4
```

```
PCASamples(meth, screeplot = TRUE)
PCASamples(meth)
dev.off()
```

```
## pdf
##   2
```

```
##############################################
#Calculate percent methylation for each sample/site
##############################################

Methylation<-getData(meth_tile)
f = function(Cyt, cov, col_name) {
  require(lazyeval)
  require(dplyr)
    mutate_call = lazyeval::interp(~ (a / b)*100, a = as.name(Cyt), b = as.name(cov))
    Methylation %>% mutate_(.dots = setNames(list(mutate_call), col_name))
}
for(i in seq_along(Index.subset$Sample.ID)){
  COVERAGE=paste0("coverage", i)
  mC=paste0("numCs", i)
  perc.mC=paste0("perc.mC_", Index.subset$Sample.ID[i])
  Methylation<-f(Cyt=mC, cov=COVERAGE, col_name=perc.mC)
}
Methylation<-dplyr::select(Methylation, chr, start, end, contains("perc.mC"))

#Merge with the percent methylation (by cytosine)
myDiff_tiles.filtered<-left_join(myDiff_tiles.filtered, Methylation)

#Subset by statistical threshold
myDiff.tiles_p05<-dplyr::filter(myDiff_tiles.filtered, pvalue<0.05)
myDiff.tiles_q05<-dplyr::filter(myDiff_tiles.filtered, qvalue<0.05)

#Save a copy of the differential Methylation analysis
wb_countData<-createWorkbook()
addWorksheet(wb_countData, "P_0.05")
  writeData(wb_countData, "P_0.05", myDiff.tiles_p05, rowNames = F)
addWorksheet(wb_countData, "Q_0.05")
  writeData(wb_countData, "Q_0.05", myDiff.tiles_q05, rowNames = F)
saveWorkbook(wb_countData, file = paste0("../2_Output/", ANALYSIS, "/DMR/", ANALYSIS, "_DiffMeth.xlsx")

myDiff.files_q05_GR<-makeGRangesFromDataFrame(myDiff.tiles_q05, seqnames.field = "chr", strand.field="s

write.table(myDiff.tiles_q05, file = "../2_Output/Tiles_Q05_DiffMeth.bed", sep = "\t", row.names = F, c

############################
##ANNOTATION
############################
library(openxlsx)
library(annotatr)
library(AnnotationHub)
library(rtracklayer)
library(TxDb.Hsapiens.UCSC.hg38.knownGene)
#convert to GRanges object (correct format for annotatr)
myDiff.tiles_p05_GR<-makeGRangesFromDataFrame(myDiff.tiles_p05, seqnames.field = "chr", strand.field="s
#create annotations from the following sources
annots = c(’hg38_cpgs’, ’hg38_basicgenes’)
# Build the annotations (a single GRanges object)
annotations = build_annotations(genome = ’hg38’, annotations = annots)
# myDiff_GR<-as(myDiff, "GRanges")
# Intersect the regions read in with the annotations
dm_annotated.tiles = annotate_regions(
    regions = myDiff.tiles_p05_GR,
    annotations = annotations,
    ignore.strand = TRUE,
    quiet = FALSE)
#convert to a data.frame
df_dm_annotated.tiles = data.frame(dm_annotated.tiles)
# A GRanges object is returned
print(dm_annotated.tiles)
```

```
## GRanges object with 158251 ranges and 8 metadata columns:
##            seqnames         ranges strand |     pvalue     qvalue meth.diff
##               <Rle>      <IRanges>  <Rle> |  <numeric>  <numeric> <numeric>
##        [1]     chr1 629501-630000      * | 3.3385e-05 0.00514958   26.0453
##        [2]     chr1 629501-630000      * | 3.3385e-05 0.00514958   26.0453
##        [3]     chr1 629501-630000      * | 3.3385e-05 0.00514958   26.0453
##        [4]     chr1 629501-630000      * | 3.3385e-05 0.00514958   26.0453
##        [5]     chr1 629501-630000      * | 3.3385e-05 0.00514958   26.0453
##        ...      ...            ...    ... .        ...        ...       ...
##   [158247]     chrM      8501-9000      * | 0.00732381   0.158793  -0.62724
##   [158248]     chrM      8501-9000      * | 0.00732381   0.158793  -0.62724
##   [158249]     chrM      8501-9000      * | 0.00732381   0.158793  -0.62724
##   [158250]     chrM      8501-9000      * | 0.00732381   0.158793  -0.62724
##   [158251]     chrM      8501-9000      * | 0.00732381   0.158793  -0.62724
##            perc.mC_1_HAEC_0.00 perc.mC_1_HAEC_2.50 perc.mC_2_HAEC_0.00
##                      <numeric>           <numeric>           <numeric>
##        [1]             74.0741             90.7216             47.2727
##        [2]             74.0741             90.7216             47.2727
##        [3]             74.0741             90.7216             47.2727
##        [4]             74.0741             90.7216             47.2727
##        [5]             74.0741             90.7216             47.2727
##        ...                 ...                 ...                 ...
##   [158247]            0.259067                   0            0.821918
##   [158248]            0.259067                   0            0.821918
##   [158249]            0.259067                   0            0.821918
##   [158250]            0.259067                   0            0.821918
##   [158251]            0.259067                   0            0.821918
##            perc.mC_2_HAEC_2.50                annot
##                      <numeric>            <GRanges>
##        [1]             62.7907 chr1:628640-629639:+
##        [2]             62.7907 chr1:626074-630073:+
##        [3]             62.7907 chr1:627757-631756:+
##        [4]             62.7907 chr1:628535-632534:+
##        [5]             62.7907 chr1:628696-632695:+
##        ...                 ...                  ...
##   [158247]                   0    chrM:6892-10891:-
##   [158248]                   0    chrM:8515-12514:-
##   [158249]                   0      chrM:8366-8572:+
##   [158250]                   0      chrM:8527-9207:+
##   [158251]                   0        chrM:1-16569:*
##   -------
##   seqinfo: 206 sequences from an unspecified genome; no seqlengths
```

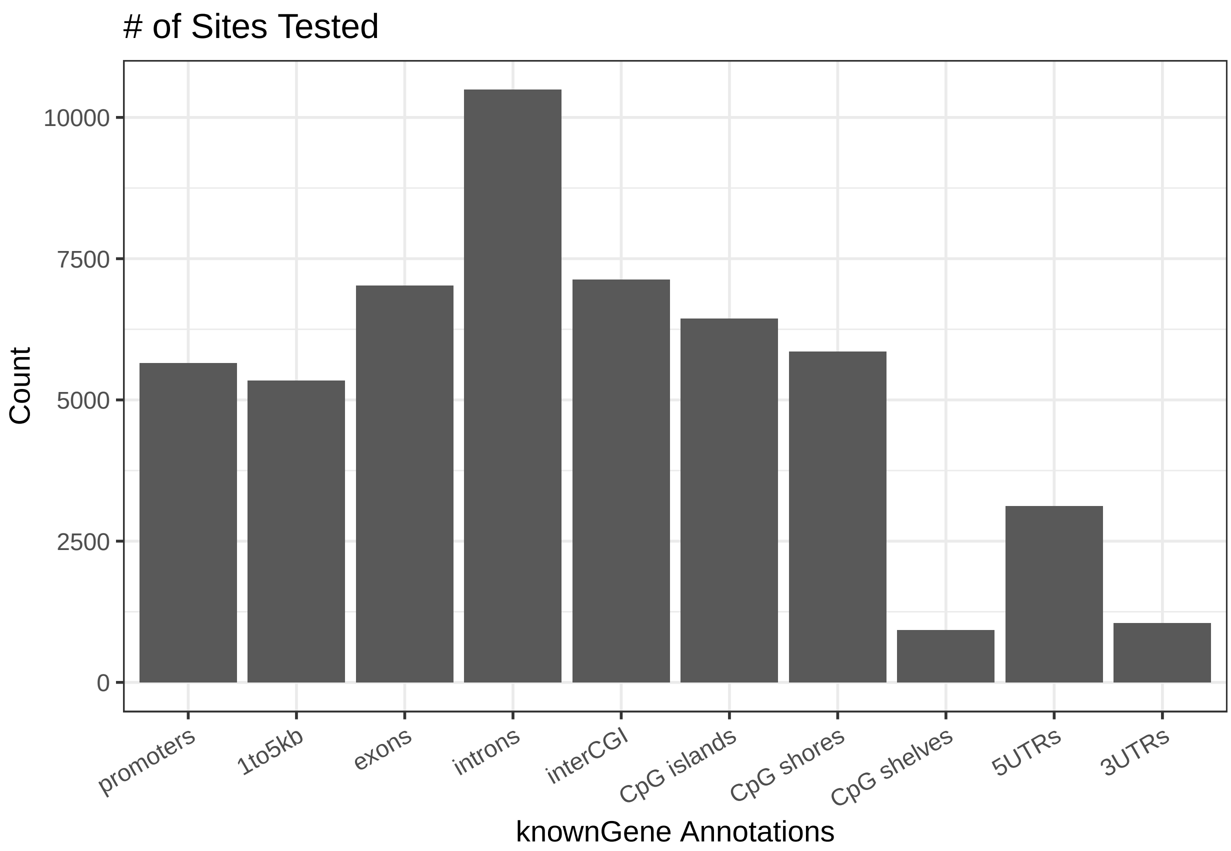
```r
##The issue with this annotation is that each DMP has multiple repeated rows if different annotations.
DiffMeth_Annotated.tiles<-df_dm_annotated.tiles %>%
  tidyr::fill(annot.symbol) %>% distinct() %>%
  dplyr::group_by(seqnames, start, end, meth.diff, pvalue, qvalue, annot.symbol) %>%
  dplyr::summarise(Annotation=paste(unique(annot.type), collapse = ";"), Test=paste(unique(annot.id), c
#Add %Methylation
DiffMeth_Annotated.tiles<-dplyr::rename(DiffMeth_Annotated.tiles, chr=seqnames)
DiffMeth_Annotated.tiles<-dplyr::left_join(DiffMeth_Annotated.tiles, Methylation)
#subset the Differential Methylation by statistics
DiffMeth_Annotated.tiles_p05<-subset(DiffMeth_Annotated.tiles, pvalue<0.05)
DiffMeth_Annotated.tiles_q05<-subset(DiffMeth_Annotated.tiles, qvalue<0.05)
#Write out the annotated DMP file
library(openxlsx)
ifelse(!dir.exists(file.path("../2_Output/", ANALYSIS, "/DMR/")), dir.create(file.path("../2_Output/", A
```

```
## [1] FALSE
```

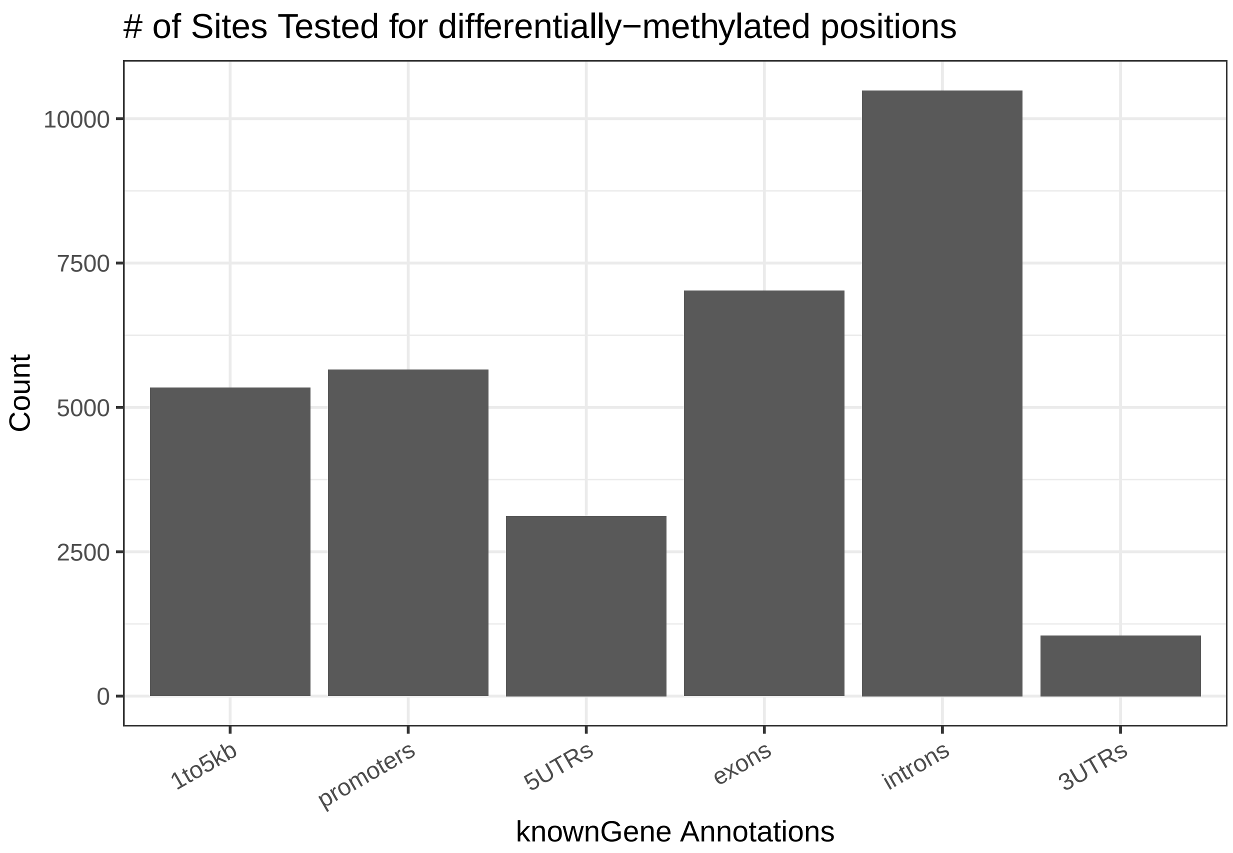
```r
wb_WGBS_Annotate<-createWorkbook()
addWorksheet(wb_WGBS_Annotate, "P_0.05")
  writeData(wb_WGBS_Annotate, "P_0.05", DiffMeth_Annotated.tiles_p05, rowNames = F)
addWorksheet(wb_WGBS_Annotate, "Q_0.05")
  writeData(wb_WGBS_Annotate, "Q_0.05", DiffMeth_Annotated.tiles_q05, rowNames = F)
saveWorkbook(wb_WGBS_Annotate, file = paste0("../2_Output/", ANALYSIS, "/DMR/", ANALYSIS, "_Annotated_D
#Provide a summary of the annotation
dm_annsum.tile = summarize_annotations(
    annotated_regions = dm_annotated.tiles,
    quiet = TRUE)
print(dm_annsum.tile)
```

```
## # A tibble: 10 x 2
##    annot.type                n
##  * <chr>                 <int>
##  1 hg38_cpg_inter         7125
##  2 hg38_cpg_islands       6439
##  3 hg38_cpg_shelves        924
##  4 hg38_cpg_shores        5850
##  5 hg38_genes_1to5kb      5339
##  6 hg38_genes_3UTRs       1051
##  7 hg38_genes_5UTRs       3118
##  8 hg38_genes_exons       7020
##  9 hg38_genes_introns    10488
## 10 hg38_genes_promoters   5651
```

```r
#Plot the annotation distribution
dm_vs_kg_annotations.tile = plot_annotation(
    annotated_regions = dm_annotated.tiles,
    plot_title = '# of Sites Tested',
    x_label = 'knownGene Annotations',
    y_label = 'Count')
print(dm_vs_kg_annotations.tile)
```



```
annots_order = c(
    'hg38_genes_1to5kb',
    'hg38_genes_promoters',
    'hg38_genes_5UTRs',
    'hg38_genes_exons',
    'hg38_genes_introns',
    'hg38_genes_3UTRs')
dm_vs_kg_annotations = plot_annotation(
    annotated_regions = dm_annotated.tiles,
    annotation_order = annots_order,
    plot_title = '# of Sites Tested for differentially-methylated positions',
    x_label = 'knownGene Annotations',
    y_label = 'Count')
print(dm_vs_kg_annotations)
```



#Heatmap of Differential Methylation

```
library(pheatmap)
hm_Data<-as.data.frame(DiffMeth_Annotated.tiles_p05)
hm_Data<-hm_Data[!is.na(hm_Data$annot.symbol),]
rownames(hm_Data)<-make.unique(hm_Data$annot.symbol, sep = ".")

##Make heatmap
STATISTIC=0.05
hm_Data<-dplyr::filter(hm_Data, pvalue<STATISTIC)

hm_Data<-dplyr::select(myDiff.tiles_p05, contains("perc.mC"))
hm_Data<-data.matrix(hm_Data)

##
##Index file for annotating samples
hm_Index<-Index.raw
hm_Index$Sample.ID<-paste0("perc.mC_", hm_Index$Sample.ID)
rownames(hm_Index)<-hm_Index$Sample.ID
hm_Index<-as.data.frame(hm_Index)
hm_Index<-dplyr::select(hm_Index, Sample.ID, Treatment)

paletteLength <- 100
myColor <- colorRampPalette(c("dodgerblue4", "white", "gold2"))(paletteLength)
pheatmap(hm_Data,
         cluster_cols=T,
         border_color=NA,
         cluster_rows=T,
         scale = ’row’,
         show_colnames = T,
         show_rownames = F,
         color = myColor,
         annotation_col = hm_Index,
         filename = paste0("../2_Output/", ANALYSIS, "/DMR/", ANALYSIS, "_Heatmap.Q05.pdf"))
```

#Figure 3: Hierarchical Clustering of Top 5 Canonical Pathways

```
library(dplyr)
library(openxlsx)
library(pheatmap)
library(RColorBrewer)
Gene.List<-read.xlsx("../1_Input/Pathway.Analysis/PANTER.Pathways.xlsx", sheet = "DMRs - Dose.Response"
Gene.List[is.na(Gene.List)]<-""
#Import datasets for the comparison
DMRs_prediabetic<-read.xlsx("../2_Output/2_Methyl/HAEC_Pre.v.CON/DMR/HAEC_Pre.v.CON_Annotated_DiffMeth.
DMRs_diabetic<-read.xlsx("../2_Output/2_Methyl/HAEC_DM.v.CON/DMR/HAEC_DM.v.CON_Annotated_DiffMeth.xlsx"

one_prediabetic<-merge(Gene.List, DMRs_prediabetic, by.x=1, by.y="annot.symbol") %>% dplyr::select(., Ge
one_diabetic<-merge(Gene.List, DMRs_diabetic, by.x=1, by.y="annot.symbol") %>% dplyr::select(., Gene=1,

two_prediabetic<-merge(Gene.List, DMRs_prediabetic, by.x=2, by.y="annot.symbol") %>% dplyr::select(., Ge
two_diabetic<-merge(Gene.List, DMRs_diabetic, by.x=2, by.y="annot.symbol") %>% dplyr::select(., Gene=1,

three_prediabetic<-merge(Gene.List, DMRs_prediabetic, by.x=3, by.y="annot.symbol") %>% dplyr::select(.,
three_diabetic<-merge(Gene.List, DMRs_diabetic, by.x=3, by.y="annot.symbol") %>% dplyr::select(., Gene=

four_prediabetic<-merge(Gene.List, DMRs_prediabetic, by.x=4, by.y="annot.symbol") %>% dplyr::select(.,
four_diabetic<-merge(Gene.List, DMRs_diabetic, by.x=4, by.y="annot.symbol") %>% dplyr::select(., Gene=1

five_prediabetic<-merge(Gene.List, DMRs_prediabetic, by.x=5, by.y="annot.symbol") %>% dplyr::select(.,
five_diabetic<-merge(Gene.List, DMRs_diabetic, by.x=5, by.y="annot.symbol") %>% dplyr::select(., Gene=1

six_prediabetic<-merge(Gene.List, DMRs_prediabetic, by.x=6, by.y="annot.symbol") %>% dplyr::select(., Ge
six_diabetic<-merge(Gene.List, DMRs_diabetic, by.x=6, by.y="annot.symbol") %>% dplyr::select(., Gene=1,

seven_prediabetic<-merge(Gene.List, DMRs_prediabetic, by.x=7, by.y="annot.symbol") %>% dplyr::select(.,
seven_diabetic<-merge(Gene.List, DMRs_diabetic, by.x=7, by.y="annot.symbol") %>% dplyr::select(., Gene=

##Combine all of the datasets
total<-full_join(one_prediabetic, one_diabetic, by=’Gene’) %>% full_join(., two_prediabetic, by=’Gene’)
row.names(total)<-total$Gene
total[is.na(total)]<-""
cluster<-dplyr::select(total, -Gene)
cluster<-data.matrix(cluster)
cluster[is.na(cluster)]<-0
##row names to display
paletteLength <- 100
myColor <- colorRampPalette(c("dodgerblue4",
    "white", "firebrick4"))(paletteLength)
```

```
# length(breaks) == length(paletteLength) + 1
# use floor and ceiling to deal with even/odd length pallettelengths
myBreaks <- c(min(cluster), seq(-50, 0, length.out=ceiling(paletteLength/2)),
              seq(2/paletteLength, 20, length.out=floor(paletteLength/2)-1),
              max(cluster))
pheatmap(cluster,
         cluster_cols=FALSE,
         border_color=NA,
         cluster_rows=TRUE,
         color = myColor,
         breaks = myBreaks,
         scale = 'none',
         show_colnames = T,
         show_rownames = T,
         fontsize=6,
         filename = "../2_Output/Hierarchical.Pathway.Analysis.pdf")
write.csv(cluster, "../2_Output/Top2.Canonical.Pathways_clustering.csv")
```

Circular Genome plot of CpG Sites

```
library(dplyr)
#create gene labels
DM.DMRs<-read.xlsx("../2_Output/2_Methyl/HAEC_DM.v.CON/DMR/HAEC_DM.v.CON_Annotated_DiffMeth.xlsx", shee
Pre.DMRs<-read.xlsx("../2_Output/2_Methyl/HAEC_Pre.v.CON/DMR/HAEC_Pre.v.CON_Annotated_DiffMeth.xlsx", sl

Gene_labels<-DM.DMRs %>% dplyr::filter(abs(meth.diff)>25) %>% dplyr::select(chrom=chr, chromStart=start
Gene_labels<-distinct(Gene_labels)
Gene_labels<-arrange(Gene_labels, chromStart)
Gene_labels$chrom<-factor(Gene_labels$chrom, levels=c("chr1", "chr2", "chr3", "chr4",
                                                      "chr5", "chr6", "chr7", "chr8",
                                                      "chr9", "chr10", "chr11", "chr12",
                                                      "chr13", "chr14", "chr15", "chr16",
                                                      "chr17", "chr18", "chr19", "chr20",
                                                      "chr21", "chr22", "chr23", "chrX",
                                                      "chrY"))
Gene_labels<-Gene_labels[order(Gene_labels$chrom),]
Gene_labels<-Gene_labels[!duplicated(Gene_labels[,4]),]
#Create gene expression
##Fold Change UP
# Gene_FoldChange.UP<-dplyr::select(DMPs_DEGs, chrom=Methyl_chr,
#                                                       chromStart=Methyl_start, FoldChange_DEG=RNA_log
# Gene_FoldChange.UP<-dplyr::filter(Gene_FoldChange.UP, FoldChange_DEG>0)
# Gene_FoldChange.UP<-dplyr::mutate(Gene_FoldChange.UP, chromEnd=chromStart+1)
# Gene_FoldChange.UP<-dplyr::select(Gene_FoldChange.UP, chrom, chromStart, chromEnd, FoldChange_DEG)
# Gene_FoldChange.UP<-arrange(Gene_FoldChange.UP, chromStart)
# Gene_FoldChange.UP$chrom<-factor(Gene_FoldChange.UP$chrom, levels=c("chr1", "chr2", "chr3", "chr4",
#                                                       "chr5", "chr6", "chr7", "chr8",
#                                                       "chr9", "chr10", "chr11", "chr12",
#                                                       "chr13", "chr14", "chr15", "chr16",
#                                                       "chr17", "chr18", "chr19", "chr20",
#                                                       "chr21", "chr22", "chr23", "chrX",
#                                                        "chrY"))
# Gene_FoldChange.UP<-Gene_FoldChange.UP[order(Gene_FoldChange.UP$chrom),]
# ##Fold Change DOWN
# Gene_FoldChange.DOWN<-dplyr::select(DMPs_DEGs, chrom=Methyl_chr,
#                                                        chromStart=Methyl_start, FoldChange_DEG=RNA_log
# Gene_FoldChange.DOWN<-dplyr::filter(Gene_FoldChange.DOWN, FoldChange_DEG<0)
# Gene_FoldChange.DOWN<-dplyr::mutate(Gene_FoldChange.DOWN, chromEnd=chromStart+1)
# Gene_FoldChange.DOWN<-dplyr::select(Gene_FoldChange.DOWN, chrom, chromStart, chromEnd, FoldChange_DEG
# Gene_FoldChange.DOWN<-arrange(Gene_FoldChange.DOWN, chromStart)
# Gene_FoldChange.DOWN$chrom<-factor(Gene_FoldChange.DOWN$chrom, levels=c("chr1", "chr2", "chr3", "chr4
#                                                        "chr5", "chr6", "chr7", "chr8",
#                                                        "chr9", "chr10", "chr11", "chr12",
#                                                        "chr13", "chr14", "chr15", "chr16",
#                                                        "chr17", "chr18", "chr19", "chr20",
#                                                        "chr21", "chr22", "chr23", "chrX",
#                                                        "chrY"))
# Gene_FoldChange.DOWN<-Gene_FoldChange.DOWN[order(Gene_FoldChange.DOWN$chrom),]
# ##Fold Change List
# Gene_FoldChange_List<-list(Gene_FoldChange.UP, Gene_FoldChange.DOWN)

# Pre Methylation Density
DMR.Pre.PerChange<-dplyr::select(Pre.DMRs, chrom=chr,
                      chromStart=start, chromEnd=end, perc.change=meth.diff)
DMR.Pre.PerChange<-DMR.Pre.PerChange[, c("chrom", "chromStart", "chromEnd", "perc.change")]
DMR.Pre.PerChange<-dplyr::filter(DMR.Pre.PerChange, chrom!="chrM")
DMR.Pre.PerChange$chrom<-factor(DMR.Pre.PerChange$chrom, levels=c("chr1", "chr2", "chr3", "chr4",
                                                      "chr5", "chr6", "chr7", "chr8",
                                                      "chr9", "chr10", "chr11", "chr12",
                                                      "chr13", "chr14", "chr15", "chr16",
                                                      "chr17", "chr18", "chr19", "chr20",
                                                      "chr21", "chr22", "chr23", "chrX",
                                                      "chrY"))
DMR.Pre.PerChange<-DMR.Pre.PerChange[order(DMR.Pre.PerChange$chrom),]
Pre.UP<-filter(DMR.Pre.PerChange, perc.change>0)
Pre.DOWN<-filter(DMR.Pre.PerChange, perc.change<0)
Pre.List<-list(Pre.DOWN, Pre.UP)

#Plot the Circos
library(circlize)
library(gtools)
library(dplyr)

circos.genomicDensity1 = function (data, ylim.force = FALSE, window.size = NULL, overlap = TRUE, col = 
data = normalizeToDataFrame(data)
if (!is.dataFrameList(data)) {
data = list(data)
}
if (length(col) == 1) {
col = rep(col, length(data))
}
if (length(lwd) == 1) {
lwd = rep(lwd, length(data))
}
if (length(lty) == 1) {
lty = rep(lty, length(data))
}
if (length(type) == 1) {
type = rep(type, length(data))
}
if (length(area) == 1) {
area = rep(area, length(data))
}

if (length(baseline) == 1) {
    baseline = rep(baseline, length(data))
}
if (length(border) == 1) {
    border = rep(border, length(data))
}
s = sapply(get.all.sector.index(), function(si) get.cell.meta.data("xrange",
    sector.index = si))

if (is.null(window.size)) {
    window.size = 10^nchar(sum(s))/1000
}
df = vector("list", length = length(data))
for (i in seq_along(data)) {
    all.chr = unique(data[[i]][[1]])
    for (chr in all.chr) {
        region = data[[i]][data[[i]][[1]] == chr, 2:3, drop = FALSE]
        dn = genomicDensity(region, window.size = window.size,
            overlap = overlap)
        dn = cbind(rep(chr, nrow(dn)), dn)
        df[[i]] = rbind(df[[i]], dn)
    }
}
if (ylim.force) {
    ymax = 1
}
else {
    ymax = max(sapply(df, function(gr) max(gr[[4]])))
}

circos.genomicTrackPlotRegion(df, ylim = c(-ymax,0), panel.fun = function(region,
    value, ...) {
    i = getI(...)

    circos.genomicLines(region, -value, col = col[i], lwd = lwd[i],
        lty = lty[i], type = type[i], border = border[i],
        area = area[i], baseline = baseline[i])
}, ...)
}

environment(circos.genomicDensity1) <- asNamespace('circlize')

#to get error line number:
```

```
f <- function (data, ylim.force = FALSE, window.size = NULL, overlap = TRUE,
col = ifelse(area, "grey", "black"), lwd = par("lwd"), lty = par("lty"),
type = "l", area = TRUE, area.baseline = NULL, baseline = 0,
border = NA, ...)
{
circos.genomicDensity1(data, ylim.force = FALSE, window.size = NULL, overlap = TRUE,
col = ifelse(area, "grey", "black"), lwd = par("lwd"), lty = par("lty"),
type = "l", area = TRUE, area.baseline = NULL, baseline = 0,
border = NA, ...)
}
###########################################################
om = circos.par("track.margin")
oc = circos.par("cell.padding")
circos.par(track.margin = c(0, 0), cell.padding = c(0, 0, 0, 0))
circos.par(start.degree = -250)
pdf(file=paste0("../2_Output/_Circos.pdf"))
circos.initializeWithIdeogram(track.height = 0.05)
### Labels for inversely changing DMRs with DEG
circos.genomicDensity(Pre.UP, col = c("coral2"), track.height = 0.1, baseline="bottom", bg.border ="whi
circos.genomicDensity(Pre.DOWN, col = c("darkcyan"), track.height = 0.1, baseline="top", bg.border ="wh
# circos.genomicLabels(Gene_labels, labels.column=4, side='inside', cex=0.6)
circos.clear()
dev.off()
```

```
## pdf
##   2
```

##Circos

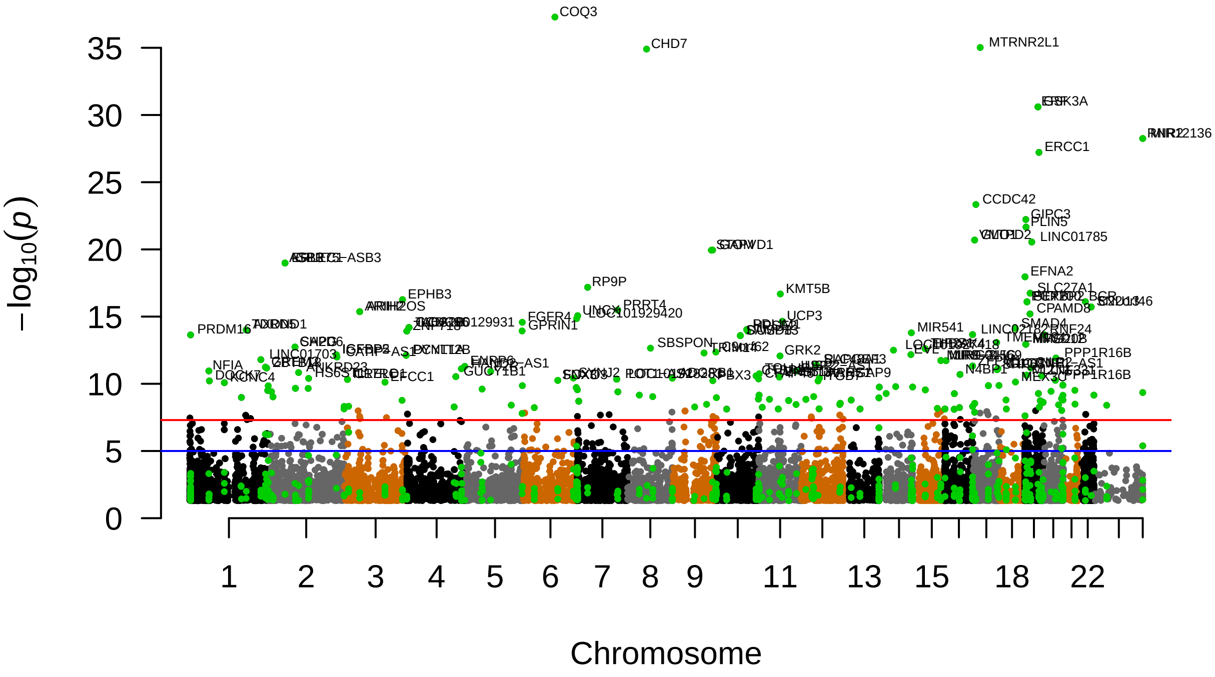
```
library(dplyr)
library(openxlsx)
Dose.DMRs<-read.xlsx("../2_Output/2_Methyl/HAEC_Dose.Response/DMR/HAEC_Dose.Response_Annotated_DiffMeth
DM.DMRs<-read.xlsx("../2_Output/2_Methyl/HAEC_DM.v.CON/DMR/HAEC_DM.v.CON_Annotated_DiffMeth.xlsx", shee
Pre.DMRs<-read.xlsx("../2_Output/2_Methyl/HAEC_Pre.v.CON/DMR/HAEC_Pre.v.CON_Annotated_DiffMeth.xlsx", s
#Gene Labels
Gene_labels<-anti_join(DM.DMRs, Pre.DMRs, by = "annot.symbol") %>% semi_join(., Dose.DMRs, by = "annot.
Gene_labels<-arrange(Gene_labels, chromStart)
Gene_labels$chrom<-factor(Gene_labels$chrom, levels=c("chr1", "chr2", "chr3", "chr4",
                                                      "chr5", "chr6", "chr7", "chr8",
                                                      "chr9", "chr10", "chr11", "chr12",
                                                      "chr13", "chr14", "chr15", "chr16",
                                                      "chr17", "chr18", "chr19", "chr20",
                                                      "chr21", "chr22", "chr23", "chrX",
                                                      "chrY"))
Gene_labels<-Gene_labels[order(Gene_labels$chrom),]
Gene_labels<-Gene_labels[!duplicated(Gene_labels[,4]),]
# DM Methylation Density
DMR.DM.PerChange<-dplyr::select(DM.DMRs, chrom=chr,
                      chromStart=start, chromEnd=end, perc.change=meth.diff)
DMR.DM.PerChange<-DMR.DM.PerChange[, c("chrom", "chromStart", "chromEnd", "perc.change")]
DMR.DM.PerChange<-dplyr::filter(DMR.DM.PerChange, chrom!="chrM")
DMR.DM.PerChange$chrom<-factor(DMR.DM.PerChange$chrom, levels=c("chr1", "chr2", "chr3", "chr4",
                                                      "chr5", "chr6", "chr7", "chr8",
                                                      "chr9", "chr10", "chr11", "chr12",
                                                      "chr13", "chr14", "chr15", "chr16",
                                                      "chr17", "chr18", "chr19", "chr20",
                                                      "chr21", "chr22", "chr23", "chrX",
                                                      "chrY"))
DMR.DM.PerChange<-DMR.DM.PerChange[order(DMR.DM.PerChange$chrom),]
DM.UP<-filter(DMR.DM.PerChange, perc.change>0)
DM.DOWN<-filter(DMR.DM.PerChange, perc.change<0)
DM.List<-list(DM.DOWN, DM.UP)

#Heatmap Data
Dose.HM.values<-dplyr::select(Dose.DMRs, annot.symbol, contains("perc.mC"))
DMR.heatmap<-anti_join(DM.DMRs, Pre.DMRs, by = "annot.symbol") %>% dplyr::select(chr, start, end, annot
DMR.HM.PerChange<-dplyr::filter(DMR.heatmap, chrom!="chrM")
DMR.HM.PerChange$chrom<-factor(DMR.HM.PerChange$chrom, levels=c("chr1", "chr2", "chr3", "chr4",
                                                      "chr5", "chr6", "chr7", "chr8",
                                                      "chr9", "chr10", "chr11", "chr12",
                                                      "chr13", "chr14", "chr15", "chr16",
                                                      "chr17", "chr18", "chr19", "chr20",
                                                      "chr21", "chr22", "chr23", "chrX",
                                                      "chrY"))
DMR.HM.PerChange<-DMR.HM.PerChange[order(DMR.HM.PerChange$chrom),]
# Pre Methylation Density
DMR.Pre.PerChange<-dplyr::select(Pre.DMRs, chrom=chr,
                      chromStart=start, chromEnd=end, perc.change=meth.diff)
DMR.Pre.PerChange<-DMR.Pre.PerChange[, c("chrom", "chromStart", "chromEnd", "perc.change")]
DMR.Pre.PerChange<-dplyr::filter(DMR.Pre.PerChange, chrom!="chrM")
DMR.Pre.PerChange$chrom<-factor(DMR.Pre.PerChange$chrom, levels=c("chr1", "chr2", "chr3", "chr4",
                                                      "chr5", "chr6", "chr7", "chr8",
                                                      "chr9", "chr10", "chr11", "chr12",
                                                      "chr13", "chr14", "chr15", "chr16",
                                                      "chr17", "chr18", "chr19", "chr20",
                                                      "chr21", "chr22", "chr23", "chrX",
                                                      "chrY"))
DMR.Pre.PerChange<-DMR.Pre.PerChange[order(DMR.Pre.PerChange$chrom),]
Pre.UP<-filter(DMR.Pre.PerChange, perc.change>0)
Pre.DOWN<-filter(DMR.Pre.PerChange, perc.change<0)
Pre.List<-list(Pre.DOWN, Pre.UP)

#Plot the Circos
library(circlize)
library(gtools)
library(dplyr)
library(RColorBrewer)

circos.genomicDensity1 = function (data, ylim.force = FALSE, window.size = NULL, overlap = TRUE, col = 
data = normalizeToDataFrame(data)
if (!is.dataFrameList(data)) {
data = list(data)
}
if (length(col) == 1) {
col = rep(col, length(data))
}
if (length(lwd) == 1) {
lwd = rep(lwd, length(data))
}
if (length(lty) == 1) {
lty = rep(lty, length(data))
}
if (length(type) == 1) {
type = rep(type, length(data))
}
if (length(area) == 1) {
area = rep(area, length(data))
}

if (length(baseline) == 1) {
    baseline = rep(baseline, length(data))
}
if (length(border) == 1) {
    border = rep(border, length(data))
}
s = sapply(get.all.sector.index(), function(si) get.cell.meta.data("xrange",
    sector.index = si))

if (is.null(window.size)) {
    window.size = 10^nchar(sum(s))/1000
}
df = vector("list", length = length(data))
for (i in seq_along(data)) {
    all.chr = unique(data[[i]][[1]])
    for (chr in all.chr) {
        region = data[[i]][data[[i]][[1]] == chr, 2:3, drop = FALSE]
        dn = genomicDensity(region, window.size = window.size,
            overlap = overlap)
        dn = cbind(rep(chr, nrow(dn)), dn)
        df[[i]] = rbind(df[[i]], dn)
    }
}
if (ylim.force) {
    ymax = 1
}
else {
    ymax = max(sapply(df, function(gr) max(gr[[4]])))
}

circos.genomicTrackPlotRegion(df, ylim = c(-ymax,0), panel.fun = function(region,
    value, ...) {
    i = getI(...)

    circos.genomicLines(region, -value, col = col[i], lwd = lwd[i],
        lty = lty[i], type = type[i], border = border[i],
        area = area[i], baseline = baseline[i])
}, ...)
}
```

```r
environment(circos.genomicDensity1) <- asNamespace(’circlize’)

#to get error line number:

f <- function (data, ylim.force = FALSE, window.size = NULL, overlap = TRUE,
col = ifelse(area, "grey", "black"), lwd = par("lwd"), lty = par("lty"),
type = "l", area = TRUE, area.baseline = NULL, baseline = 0,
border = NA, ...)
{
circos.genomicDensity1(data, ylim.force = FALSE, window.size = NULL, overlap = TRUE,
col = ifelse(area, "grey", "black"), lwd = par("lwd"), lty = par("lty"),
type = "l", area = TRUE, area.baseline = NULL, baseline = 0,
border = NA, ...)
}

om = circos.par("track.margin")
oc = circos.par("cell.padding")
circos.par(track.margin = c(0, 0), cell.padding = c(0, 0, 0, 0))
circos.par(gap.after = 2, start.degree = -250)
pdf(file=paste0("../2_Output/Circos.pdf"), width=8.5, height=8.5)
circos.initializeWithIdeogram(plotType = NULL)
# Labels
circos.genomicLabels(Gene_labels, labels.column = 4, side = "outside")
### DM Density
circos.genomicDensity(DMR.DM.PerChange, col = c("brown2"), track.height = 0.1, baseline="bottom", bg.bo
# circos.genomicDensity1(DM.DOWN, col = c("darkblue"), track.height = 0.1, baseline="top", bg.border ="
### Pre Density
# circos.genomicDensity(DMR.Pre.PerChange, col = c("coral2"), track.height = 0.1, baseline="bottom", bg
# circos.genomicDensity1(Pre.DOWN, col = c("darkcyan"), track.height = 0.1, baseline="top", bg.border =
##DMRs in DM
circos.genomicTrackPlotRegion(DM.List,
                              ylim = c(-75, 75), bg.border=NA,
                              panel.fun = function(region, value, ...) {
 col = ifelse(value[[1]] > 0, "darkgoldenrod1", "dodgerblue2")
 circos.genomicPoints(region, value, col = add_transparency(col, 0.2), cex = 0.5, pch = 16)
 cell.xlim = get.cell.meta.data("cell.xlim")
 for(h in c(-75, -50, -25, 0, 25, 50, 75)) {
   circos.lines(cell.xlim, c(h, h), col ="#00000040")
 }
}, track.height = 0.2)
#Heatmap
col_fun = colorRamp2(c(0, 50, 100), c("darkcyan", "white", "coral2"))
circos.genomicHeatmap(DMR.HM.PerChange, col = col_fun, side = "inside", border = "white")

circos.genomicIdeogram()
circos.clear()
dev.off()
```

```
## pdf
##   2
```

#DMP Distribution: Manhattan plot

```r
library(qqman)
man_data<-dplyr::rename(DiffMeth_Annotated.tiles_p05, CHR=chr, BP=start, P=pvalue, SNP=annot.symbol)
man_data$CHR<-sub("chr","", man_data$CHR)
man_data<-dplyr::filter(man_data, CHR!="Y")
man_data$CHR<-gsub("X", "23", man_data$CHR)
man_data$CHR<-gsub("M", "25", man_data$CHR)
man_data$CHR<-as.numeric(man_data$CHR)
ANNOTATE<-unique(filter(man_data, P<0.00000001)$SNP)
manhattan(man_data, chr = "CHR", bp = "BP", p = "P", snp = "SNP",
      col = c("black","#666666","#CC6600"), chrlabs = c(1:22, "X", "MT"),
      suggestiveline = -log10(1e-05), genomewideline = -log10(5e-08),
      highlight = ANNOTATE, logp = TRUE, annotatePval = 1e-10, annotateTop = F, cex=0.5)
```



```r
#Make PDF
pdf(file = paste0("../2_Output/", ANALYSIS, "/", ANALYSIS, "_Manhattan.pdf"), height=3, width=5)
manhattan(man_data, chr = "CHR", bp = "BP", p = "P", snp = "SNP",
      col = c("black","#666666","#CC6600"), chrlabs = c(1:22, "X", "MT"),
      suggestiveline = -log10(1e-8), genomewideline = -log10(5e-10),
      logp = TRUE, annotatePval = 1e-08, annotateTop = F, cex=0.5)
dev.off()
```

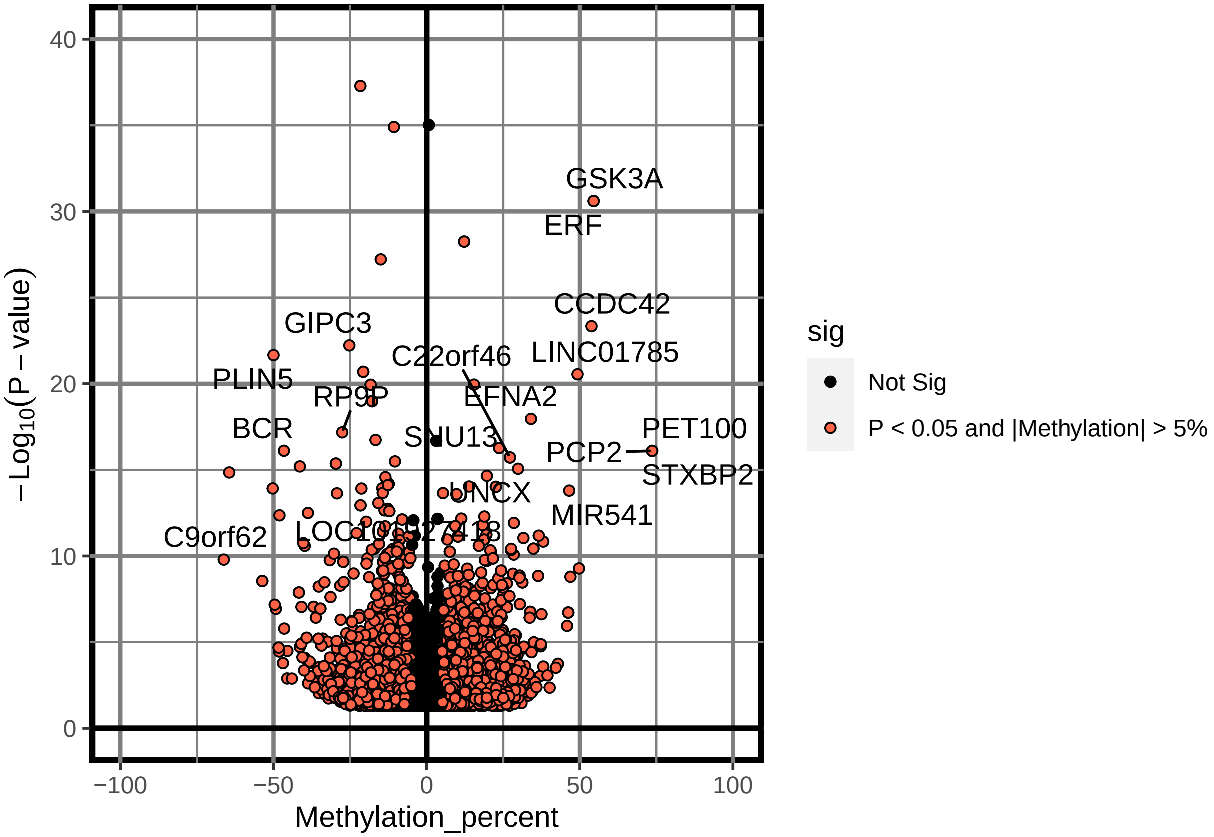
```
## pdf
##   2
```

##Volcano Plot

```r
# Load packages
library(dplyr)
library(ggplot2)
library(ggrepel)
library(openxlsx)
# Read data from the web
results<-DiffMeth_Annotated.tiles_p05
results <- results %>% dplyr::select(annot.symbol,pvalue, meth.diff) %>% mutate(minuslogpvalue=-log10(p
results$sig<-ifelse(results$pvalue<0.05 & abs(results$meth.diff)>5, "P < 0.05 and |Methylation| > 5%",
max(results$minuslogpvalue, na.rm = TRUE)
```

```
## [1] 37.28479
```

```r
#plot the ggplot
p = ggplot(results, aes(meth.diff, minuslogpvalue)) + theme(panel.background = element_rect("white", col
geom_point(aes(fill=sig), colour="black", shape=21) + labs(x=expression(Methylation_percent), y=express
scale_fill_manual(values=c("black", "tomato"))
#add a repelling effect to the text labels.
p+geom_text_repel(data=filter(results, minuslogpvalue>10 & abs(meth.diff)>25 | minuslogpvalue>7 & abs(me
```



```r
pdf(file = paste0("../2_Output/", ANALYSIS, "/", ANALYSIS, "Volcano.Plot.pdf"))
p+geom_text_repel(data=filter(results, minuslogpvalue>10 & abs(meth.diff)>25 | minuslogpvalue>7 & abs(m
dev.off()
```

```
## pdf
##   2
```

#Methylation Distribution using **EnrichedHeatmap**

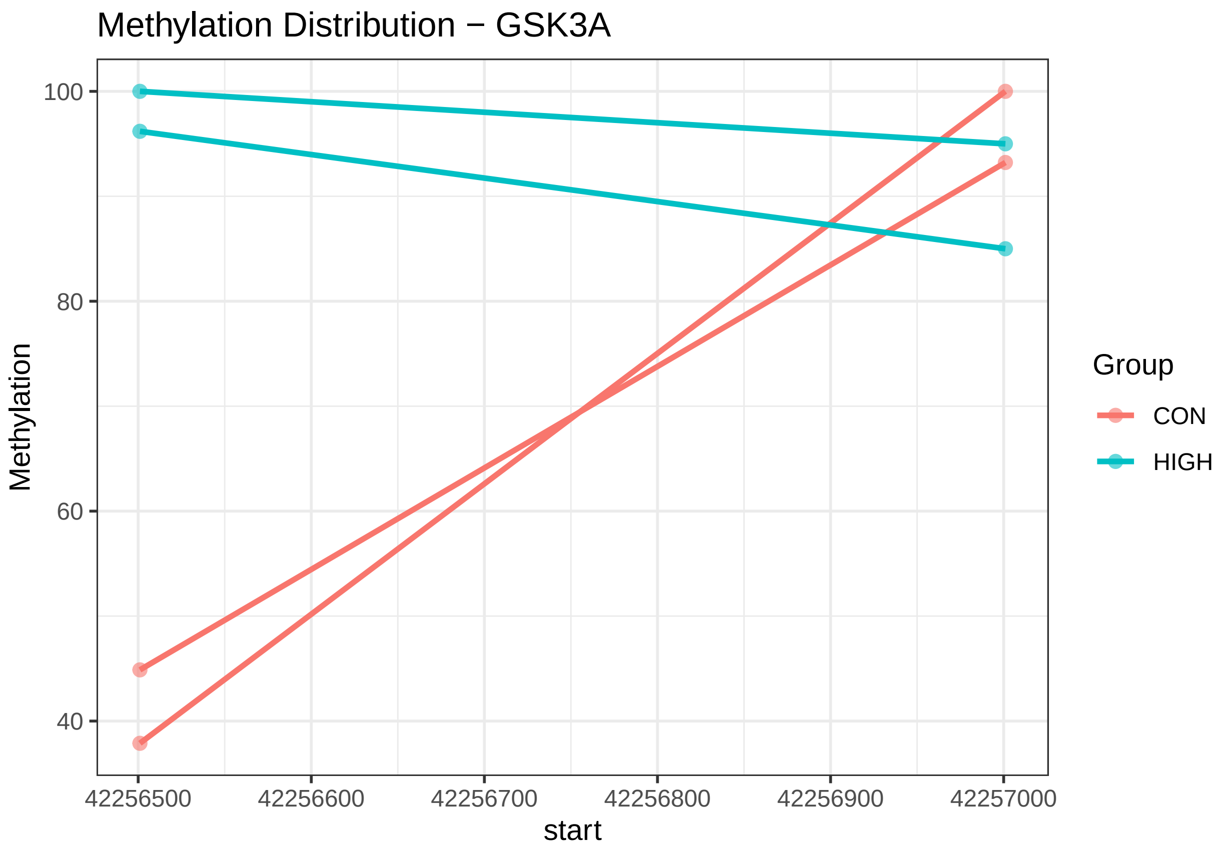
```
#Import the genomic annotation file
library(EnrichedHeatmap)
library(annotatr)
library(TxDb.Hsapiens.UCSC.hg38.knownGene)
annots = c("hg38_basicgenes", "hg38_genes_promoters", "hg38_genes_intergenic",
           "hg38_genes_intronexonboundaries", "hg38_cpgs", "hg38_cpg_islands", "hg38_cpg_shores", "hg38_
annotations=build_annotations(genome = "hg38", annotations = annots)
annotations<-keepStandardChromosomes(annotations, pruning.mode = "coarse") #Remove nonstandard chromoso

myDiff.tiles_p05<-dplyr::mutate(myDiff.tiles_p05, absolute.meth=abs(meth.diff))
myDiff.tiles_p05<-as(myDiff.tiles_p05, "GRanges")

# #Import the annotated "target" data
# myDiff<-openxlsx::read.xlsx(paste0("../2_Input/WGBS_MethylKit_DiffMeth", ANALYSIS,".xlsx"), sheet = "
# myDiff_GR<-as(myDiff, "GRanges")

#Annotate GRanges using hg38 genome
dm_annotated = annotate_regions(
  regions = myDiff.tiles_p05,
  annotations = annotations,
  ignore.strand = TRUE)
#create data.frame
df_dm_annotated <- as.data.frame(dm_annotated)
#
library(GenomicFeatures)
genes<-genes(TxDb.Hsapiens.UCSC.hg38.knownGene)
tss = promoters(genes, upstream = 0, downstream = 1)
mat1 = normalizeToMatrix(myDiff.tiles_p05, tss, value_column = "absolute.meth", extend = 5000, mean_mode
EnrichedHeatmap(mat1, col = c("white", "black"), name = ANALYSIS)
#
# png(file = paste0("../2_Output/2_Methyl/", ANALYSIS, "/","_1Methyl.Gene.Distribution.png"), height = 
# EnrichedHeatmap(mat1, col = c("white", "black"), name = "Heart Failure")
# dev.off()

partition = kmeans(mat1, centers = 3)$cluster
lgd = Legend(at = c("cluster1", "cluster2", "cluster3"), title = "Clusters",
    type = "lines", legend_gp = gpar(col = 2:4))
ht_list = Heatmap(partition, col = structure(2:4, names = as.character(1:3)), name = "partition",
              show_row_names = FALSE, width = unit(3, "mm")) + EnrichedHeatmap(mat1, col = c("white", "
draw(ht_list, main_heatmap = "|PercentMethylation|")

partition = kmeans(mat1, centers = 2)$cluster
lgd = Legend(at = c("cluster1", "cluster2"), title = "Clusters",
    type = "lines", legend_gp = gpar(col = 2:3))
ht_list = Heatmap(partition, col = structure(2:3, names = as.character(1:2)), name = "partition",
              show_row_names = FALSE, width = unit(3, "mm")) + EnrichedHeatmap(mat1, col = c("white", "
pdf(file = paste0("../2_Output/2_Methyl/", ANALYSIS, "/", ANALYSIS, "_Methyl.Gene.Distribution_Kmeans.p
draw(ht_list, main_heatmap = "|PercentMethylation|")
dev.off()
```

```
## pdf
##   2
```

```
# library
library(ggridges)
library(ggplot2)
DMPs<-DiffMeth_Annotated.tiles_p05
colData<-Index.subset
GENE="GSK3A"
Gene_DMP<-dplyr::filter(DMPs, annot.symbol %in% GENE)
Gene_DMP<-dplyr::select(Gene_DMP, chr,start, end, annot.symbol, contains("perc.mC"))
colnames(Gene_DMP)<-gsub("perc.mC_", "", colnames(Gene_DMP))
gathered<-tidyr::gather(Gene_DMP, "Sample", "Methylation", 8:length(colnames(Gene_DMP)))
gathered_annot<-merge(colData, gathered, by.x = "Sample.ID", by.y = "Sample")
gathered_annot$Methylation<-as.numeric(as.character(gathered_annot$Methylation))
gathered_annot$Group<-factor(gathered_annot$Group, levels = c("CON", "PRE", "HIGH"))
# basic example
pdf(file=paste0("../2_Output/", ANALYSIS, "/", ANALYSIS, "_", GENE, "_Methylation_gene.distribution.pdf
ggplot(gathered_annot, aes(x=start, y=Methylation, group = Sample.ID, color=Group))+theme_bw()+geom_lin
dev.off()
```

```
## pdf
##   2
```

```
ggplot(gathered_annot, aes(x=start, y=Methylation, group = Sample.ID, color=Group))+theme_bw()+geom_lin
```



#Supplemental Table: R Session Information

All packages and setting are acquired using the following command:

```r
sinfo<-devtools::session_info()
sinfo$platform
```

```
##  setting  value
##  version  R version 4.0.3 (2020-10-10)
##  os       macOS Big Sur 10.16
##  system   x86_64, darwin17.0
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  ctype    en_US.UTF-8
##  tz       Europe/Berlin
##  date     2021-01-21
```

```r
sinfo$packages %>% kable(
                         align="c",
                         longtable=T,
                         booktabs=T,
                         caption="Packages and Required Dependencies") %>%
    kable_styling(latex_options=c("striped", "repeat_header", "condensed"))
```

	package	ondiskversion	loadedversion
AnnotationDbi	AnnotationDbi	1.52.0	1.52.0
AnnotationHub	AnnotationHub	2.22.0	2.22.0
annotatr	annotatr	1.16.0	1.16.0
askpass	askpass	1.1	1.1
assertthat	assertthat	0.2.1	0.2.1
bbmle	bbmle	1.0.23.1	1.0.23.1
bdsmatrix	bdsmatrix	1.3.4	1.3-4
Biobase	Biobase	2.50.0	2.50.0
BiocFileCache	BiocFileCache	1.14.0	1.14.0
BiocGenerics	BiocGenerics	0.36.0	0.36.0
BiocManager	BiocManager	1.30.10	1.30.10
BiocParallel	BiocParallel	1.24.1	1.24.1
BiocVersion	BiocVersion	3.12.0	3.12.0
biomaRt	biomaRt	2.46.0	2.46.0
Biostrings	Biostrings	2.58.0	2.58.0
bit	bit	4.0.4	4.0.4
bit64	bit64	4.0.5	4.0.5
bitops	bitops	1.0.6	1.0-6
blob	blob	1.2.1	1.2.1
BSgenome	BSgenome	1.58.0	1.58.0
Cairo	Cairo	1.5.12.2	1.5-12.2
calibrate	calibrate	1.7.7	1.7.7
callr	callr	3.5.1	3.5.1

	package	ondiskversion	loadedversion
circlize	circlize	0.4.12	0.4.12
cli	cli	2.2.0	2.2.0
clue	clue	0.3.58	0.3-58
cluster	cluster	2.1.0	2.1.0
coda	coda	0.19.4	0.19-4
codetools	codetools	0.2.18	0.2-18
colorspace	colorspace	2.0.0	2.0-0
ComplexHeatmap	ComplexHeatmap	2.6.2	2.6.2
crayon	crayon	1.3.4	1.3.4
curl	curl	4.3	4.3
data.table	data.table	1.13.6	1.13.6
DBI	DBI	1.1.1	1.1.1
dbplyr	dbplyr	2.0.0	2.0.0
DelayedArray	DelayedArray	0.16.0	0.16.0
desc	desc	1.2.0	1.2.0
devtools	devtools	2.3.2	2.3.2
digest	digest	0.6.27	0.6.27
dplyr	dplyr	1.0.3	1.0.3
ellipsis	ellipsis	0.3.1	0.3.1
emdbook	emdbook	1.3.12	1.3.12
EnrichedHeatmap	EnrichedHeatmap	1.20.0	1.20.0
evaluate	evaluate	0.14	0.14
fansi	fansi	0.4.2	0.4.2
farver	farver	2.0.3	2.0.3
fastmap	fastmap	1.0.1	1.0.1
fastseg	fastseg	1.36.0	1.36.0
fs	fs	1.5.0	1.5.0
generics	generics	0.1.0	0.1.0
GenomeInfoDb	GenomeInfoDb	1.26.2	1.26.2
GenomeInfoDbData	GenomeInfoDbData	1.2.4	1.2.4
GenomicAlignments	GenomicAlignments	1.26.0	1.26.0
GenomicFeatures	GenomicFeatures	1.42.1	1.42.1
GenomicRanges	GenomicRanges	1.42.0	1.42.0
GetoptLong	GetoptLong	1.0.5	1.0.5
ggplot2	ggplot2	3.3.3	3.3.3
ggrepel	ggrepel	0.9.1	0.9.1
ggridges	ggridges	0.5.3	0.5.3
GlobalOptions	GlobalOptions	0.1.2	0.1.2
glue	glue	1.4.2	1.4.2
gtable	gtable	0.3.0	0.3.0
gtools	gtools	3.8.2	3.8.2
hms	hms	1.0.0	1.0.0
htmltools	htmltools	0.5.1	0.5.1
httpuv	httpuv	1.5.5	1.5.5
httr	httr	1.4.2	1.4.2
interactiveDisplayBase	interactiveDisplayBase	1.28.0	1.28.0
IRanges	IRanges	2.24.1	2.24.1

	package	ondiskversion	loadedversion
kableExtra	kableExtra	1.3.1	1.3.1
knitr	knitr	1.30	1.30
labeling	labeling	0.4.2	0.4.2
later	later	1.1.0.1	1.1.0.1
lattice	lattice	0.20.41	0.20-41
lazyeval	lazyeval	0.2.2	0.2.2
lifecycle	lifecycle	0.2.0	0.2.0
limma	limma	3.46.0	3.46.0
locfit	locfit	1.5.9.4	1.5-9.4
magrittr	magrittr	2.0.1	2.0.1
MASS	MASS	7.3.53	7.3-53
Matrix	Matrix	1.3.2	1.3-2
MatrixGenerics	MatrixGenerics	1.2.0	1.2.0
matrixStats	matrixStats	0.57.0	0.57.0
mclust	mclust	5.4.7	5.4.7
memoise	memoise	1.1.0	1.1.0
methylKit	methylKit	1.16.0	1.16.0
mgcv	mgcv	1.8.33	1.8-33
mime	mime	0.9	0.9
munsell	munsell	0.5.0	0.5.0
mvtnorm	mvtnorm	1.1.1	1.1-1
nlme	nlme	3.1.151	3.1-151
numDeriv	numDeriv	2016.8.1.1	2016.8-1.1
openssl	openssl	1.4.3	1.4.3
openxlsx	openxlsx	4.2.3	4.2.3
org.Hs.eg.db	org.Hs.eg.db	3.12.0	3.12.0
pheatmap	pheatmap	1.0.12	1.0.12
pillar	pillar	1.4.7	1.4.7
pkgbuild	pkgbuild	1.2.0	1.2.0
pkgconfig	pkgconfig	2.0.3	2.0.3
pkgload	pkgload	1.1.0	1.1.0
plyr	plyr	1.8.6	1.8.6
png	png	0.1.7	0.1-7
prettyunits	prettyunits	1.1.1	1.1.1
processx	processx	3.4.5	3.4.5
progress	progress	1.2.2	1.2.2
promises	promises	1.1.1	1.1.1
ps	ps	1.5.0	1.5.0
purrr	purrr	0.3.4	0.3.4
qqman	qqman	0.1.4	0.1.4
qvalue	qvalue	2.22.0	2.22.0
R.methodsS3	R.methodsS3	1.8.1	1.8.1
R.oo	R.oo	1.24.0	1.24.0
R.utils	R.utils	2.10.1	2.10.1
R6	R6	2.5.0	2.5.0
rappdirs	rappdirs	0.3.1	0.3.1

	package	ondiskversion	loadedversion	
RColorBrewer	RColorBrewer	1.1.2	1.1-2	
Rcpp	Rcpp	1.0.6	1.0.6	
RCurl	RCurl	1.98.1.2	1.98-1.2	
readr	readr	1.4.0	1.4.0	
regioneR	regioneR	1.22.0	1.22.0	
remotes	remotes	2.2.0	2.2.0	
reshape2	reshape2	1.4.4	1.4.4	
rjson	rjson	0.2.20	0.2.20	
rlang	rlang	0.4.10	0.4.10	
rmarkdown	rmarkdown	2.6	2.6	
rprojroot	rprojroot	2.0.2	2.0.2	
Rsamtools	Rsamtools	2.6.0	2.6.0	
RSQLite	RSQLite	2.2.2	2.2.2	
rstudioapi	rstudioapi	0.13	0.13	
rtracklayer	rtracklayer	1.50.0	1.50.0	
rvest	rvest	0.3.6	0.3.6	
S4Vectors	S4Vectors	0.28.1	0.28.1	
scales	scales	1.1.1	1.1.1	
sessioninfo	sessioninfo	1.1.1	1.1.1	
shape	shape	1.4.5	1.4.5	
shiny	shiny	1.5.0	1.5.0	
stringi	stringi	1.5.3	1.5.3	
stringr	stringr	1.4.0	1.4.0	
SummarizedExperiment	SummarizedExperiment	1.20.0	1.20.0	
testthat	testthat	3.0.1	3.0.1	
tibble	tibble	3.0.5	3.0.5	
tidyr	tidyr	1.1.2	1.1.2	
tidyselect	tidyselect	1.1.0	1.1.0	
TxDb.Hsapiens.UCSC.hg38.knownGene	TxDb.Hsapiens.UCSC.hg38.knownGene	3.10.0	3.10.0	/Libr
usethis	usethis	2.0.0	2.0.0	
utf8	utf8	1.1.4	1.1.4	
vctrs	vctrs	0.3.6	0.3.6	
viridisLite	viridisLite	0.3.0	0.3.0	
webshot	webshot	0.5.2	0.5.2	
withr	withr	2.4.0	2.4.0	
xfun	xfun	0.20	0.20	
XML	XML	3.99.0.5	3.99-0.5	
xml2	xml2	1.3.2	1.3.2	
xtable	xtable	1.8.4	1.8-4	
XVector	XVector	0.30.0	0.30.0	
yaml	yaml	2.2.1	2.2.1	
zip	zip	2.1.1	2.1.1	
zlibbioc	zlibbioc	1.36.0	1.36.0	