Epigenomics of ASCs - WGBS Analysis Pipeline

Mark E. Pepin

02/05/2020

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 ##Genome Assembly The first task was to align the bisulfite reduced and sequenced reads to a genome assembly. To accomplish this, an annotated human genome assembly was prepared based on Gencode annotation (gencode.v28.annotation.gtf) and sequence (GRCh38.p12.genome.fa). `bwameth.py index /data/scratch/pepinme/Napoli/Input/Genome/GRCh38.p12.genome.fa`

The sequencing data were then trimmed and aligned using the following batch script: ##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 All .fastq files were then aligned 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_001_val_1.fq.gz $INPUT_DIR/fastq_trimmed/${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 > $RESULTS_DIR/RRBS_bwa/${VA`
###Sort using samtools `samtools sort $RESULTS_DIR/RRBS_bwa/${VAR}.bam -o $RESULTS_DIR/RRBS_bwa/${VAR}.sor`
###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 (version 3.6.1) and combined into a single “object” for differential methylation analysis

#Differential Methylation Analysis

##Combining sample methylation

```r
#Conditions to be used in differential methylation analysis (FILL OUT)
library(openxlsx)
library(dplyr)
TREATMENT=c("VEH","AZA")
CELL=c("ASC")

ANALYSIS="ASC_AZAvCON"
### "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$Treatment<-factor(Index.raw$Treatment, levels = c("VEH", "AZA"))
# 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<-Index.raw
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)
```

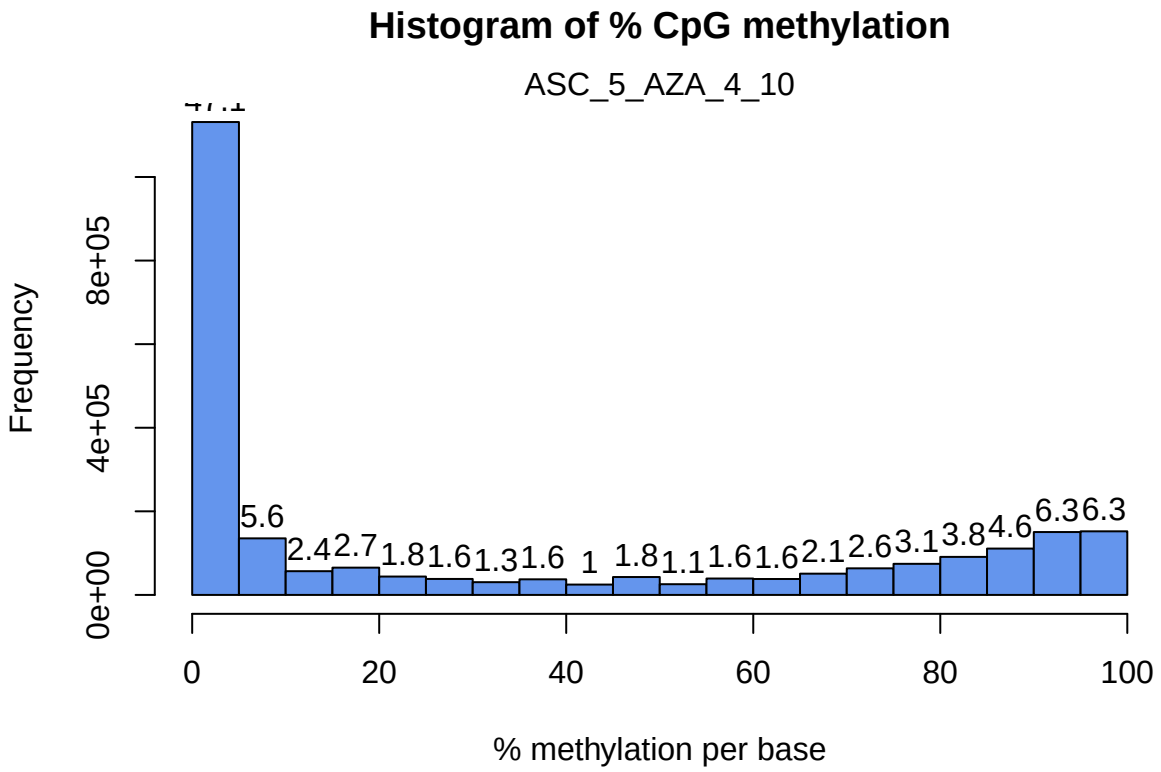
```
## methylation statistics per base
## summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   7.692  32.142  73.913 100.000
## percentiles:
##         0%        10%        20%        30%        40%        50%        60%
##   0.000000   0.000000   0.000000   0.000000   0.000000   7.692308  26.666667
##        70%        80%        90%        95%        99%      99.5%      99.9%
##  61.904762  81.818182  92.307692 100.000000 100.000000 100.000000 100.000000
##       100%
## 100.000000
```

```r
#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)
```

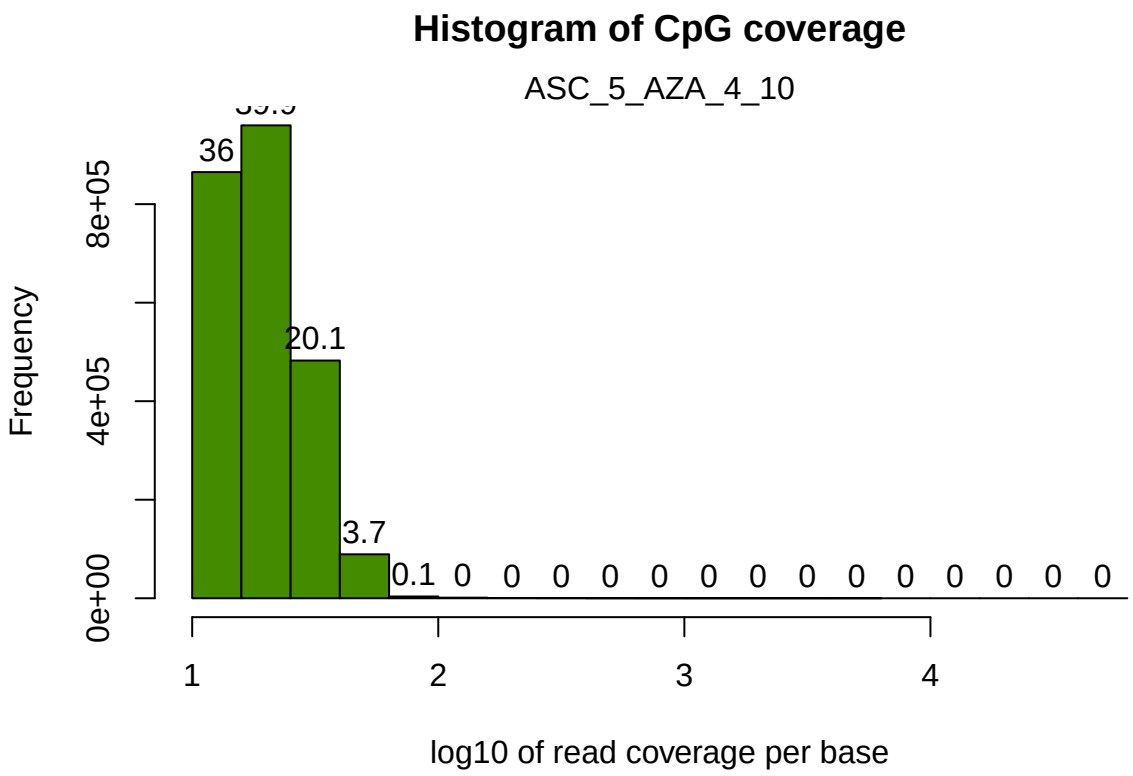
Once the samples have been compiled, it is valuable to perform some basic visualizations and statistics to determine whether quality filtering is necessary. The distribution of methylation change is plotted as a histogram (typically bimodal at the extremes), as well as a distribution of the read coverage 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, however, coverage distribution exhibits a one-tailed distribution, lowering concern that the analysis is confounded by PCR amplification bias in coverage.

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




```
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.per
meth<-unite(filtered.myobj, destrand = FALSE) #When calculating DMRs, it is not helpful to "destrand"
```

##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 = "F", mc.cores = 7)
myDiff_tile.md<-as(myDiff_tiles,"methylDiff")
myDiff_tiles.filtered<-dplyr::select(myDiff_tile.md, chr, start, end, strand, meth.diff, pvalue, qvalue)
###############################################
#Calculate percent methylation for each sample/site
###############################################
Methylation<-as.data.frame(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

############################
```

```r
##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 178960 ranges and 8 metadata columns:
##            seqnames        ranges strand |         meth.diff
##               <Rle>     <IRanges>  <Rle> |         <numeric>
##        [1]     chr1 631501-632000      * | -35.8122038673326
##        [2]     chr1 631501-632000      * | -35.8122038673326
##        [3]     chr1 631501-632000      * | -35.8122038673326
##        [4]     chr1 631501-632000      * | -35.8122038673326
##        [5]     chr1 631501-632000      * | -35.8122038673326
##        ...      ...           ...    ... .               ...
##   [178956]     chrM   13501-14000      * | 0.906926950634562
##   [178957]     chrM   13501-14000      * | 0.906926950634562
##   [178958]     chrM   13501-14000      * | 0.906926950634562
##   [178959]     chrM   13501-14000      * | 0.906926950634562
##   [178960]     chrM   13501-14000      * | 0.906926950634562
##                         pvalue              qvalue perc.mC_ASC_5_AZA_12_10
##                      <numeric>           <numeric>              <numeric>
##        [1] 1.0951176040696e-24 1.38776510707954e-20        8.44629822732012
##        [2] 1.0951176040696e-24 1.38776510707954e-20        8.44629822732012
##        [3] 1.0951176040696e-24 1.38776510707954e-20        8.44629822732012
##        [4] 1.0951176040696e-24 1.38776510707954e-20        8.44629822732012
##        [5] 1.0951176040696e-24 1.38776510707954e-20        8.44629822732012
##        ...                 ...                 ...                     ...
##   [178956]  0.0296664492352993    0.220542309822812        2.47578040904198
##   [178957]  0.0296664492352993    0.220542309822812        2.47578040904198
##   [178958]  0.0296664492352993    0.220542309822812        2.47578040904198
##   [178959]  0.0296664492352993    0.220542309822812        2.47578040904198
##   [178960]  0.0296664492352993    0.220542309822812        2.47578040904198
##            perc.mC_ASC_5_AZA_4_10 perc.mC_ASC_DMSO_12_10 perc.mC_ASC_DMSO_4_10
```

```
##                         <numeric>              <numeric>            <numeric>
##         [1]        64.963503649635                      0      72.2772277227723
##         [2]        64.963503649635                      0      72.2772277227723
##         [3]        64.963503649635                      0      72.2772277227723
##         [4]        64.963503649635                      0      72.2772277227723
##         [5]        64.963503649635                      0      72.2772277227723
##         ...                    ...                    ...                   ...
##    [178956]       2.63157894736842       1.62818235642392      1.65837479270315
##    [178957]       2.63157894736842       1.62818235642392      1.65837479270315
##    [178958]       2.63157894736842       1.62818235642392      1.65837479270315
##    [178959]       2.63157894736842       1.62818235642392      1.65837479270315
##    [178960]       2.63157894736842       1.62818235642392      1.65837479270315
##                           annot
##                       <GRanges>
##         [1] chr1:631757-632756:+
##         [2] chr1:631205-632204:-
##         [3] chr1:631151-632150:-
##         [4] chr1:627757-631756:+
##         [5] chr1:628535-632534:+
##         ...                  ...
##    [178956]     chrM:-3398-601:+
##    [178957]     chrM:-3329-670:+
##    [178958]    chrM:9747-13746:+
##    [178959]   chrM:10888-14887:+
##    [178960]   chrM:12337-14148:+
##    -------
##    seqinfo: 229 sequences from an unspecified genome; no seqlengths
```

```
##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/",
```

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

```
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)
```

#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"))
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)
```

Circular Genome plot of CpG Sites

```
library(dplyr)
library(tidyr)
#create gene labels
Gene_labels<-DiffMeth_Annotated.tiles_p05 %>% dplyr::filter(abs(meth.diff)>10, -log(qvalue)>8, grepl("p
Gene_labels<-distinct(Gene_labels)
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 %>% group_by(chrom) %>% arrange(chrom, chromStart)
Gene_labels<-Gene_labels[!duplicated(Gene_labels[,"GeneSymbol"]),]
Gene_labels<-Gene_labels[c("chrom", "chromStart", "chromEnd", "GeneSymbol", "meth.diff", "pvalue", "qval
Gene_labels<-dplyr::filter(Gene_labels, GeneSymbol!="NA")
Gene_labels<-as.data.frame(Gene_labels)
Gene_labels<-Gene_labels[complete.cases(Gene_labels),]
# Methylation Density
DMR.PerChange<-as.data.frame(dplyr::mutate(DiffMeth_Annotated.tiles_p05, chrom=chr,
                      chromStart=as.integer(start), chromEnd=as.integer(chromStart+500), perc.change=me
DMR.PerChange<-DMR.PerChange[c("chrom", "chromStart", "chromEnd", "perc.change")]
DMR.PerChange<-dplyr::filter(DMR.PerChange, chrom!="chrM")
DMR.PerChange$chrom<-factor(DMR.PerChange$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.PerChange<-DMR.PerChange[order(DMR.PerChange$chromStart),]
DMR.PerChange<-DMR.PerChange[order(DMR.PerChange$chrom),]
Methyl.UP<-filter(DMR.PerChange, perc.change>0)
Methyl.DOWN<-filter(DMR.PerChange, perc.change<0)
Methyl.List<-list(Methyl.DOWN, Methyl.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) {
```

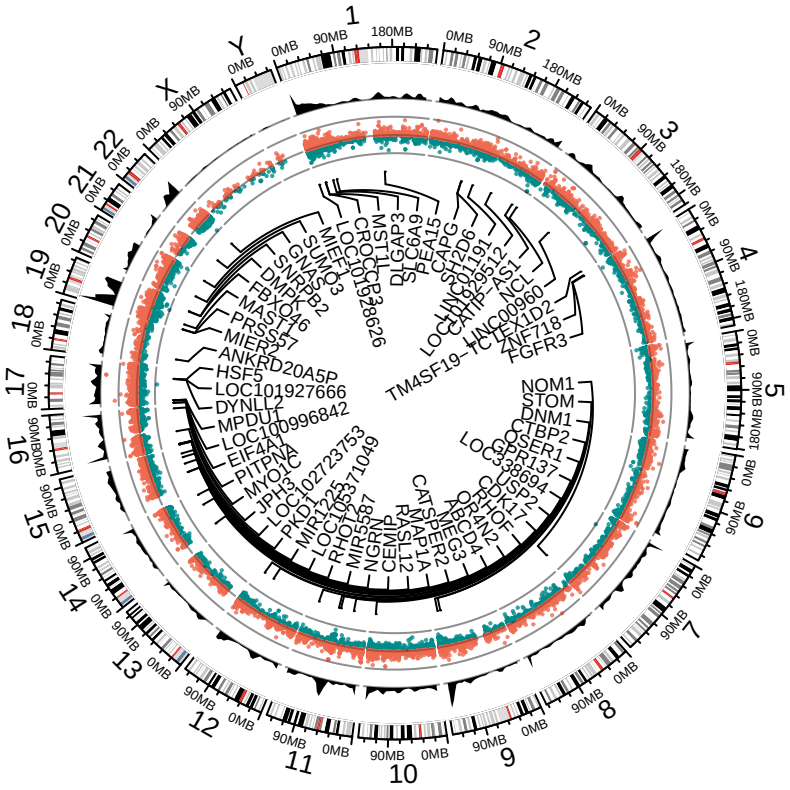
```
lty = rep(lty, length(data))
}
if (length(type) == 1) {
type = rep(type, length(data))
}
if (length(area) == 1) {
area = rep(area, length(data))
}

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, ...)
}
#Create Circos
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/", ANALYSIS, "/", ANALYSIS, "_Circos.pdf"))
circos.initializeWithIdeogram(track.height = 0.05)
### Labels for inversely changing DMRs with DEG
circos.genomicDensity(DMR.PerChange, col = c("black"), track.height = 0.1, baseline="bottom", bg.border
##DEG with inverse GPI Islands Promoters
circos.genomicTrackPlotRegion(Methyl.List,
                              ylim = c(-100, 100), bg.border=NA,
                              panel.fun = function(region, value, ...) {
 col = ifelse(value[[1]] > 0, "coral2", "darkcyan")
 circos.genomicPoints(region, value, col = add_transparency(col, 0.2), cex = 0.3, pch = 16)
 cell.xlim = get.cell.meta.data("cell.xlim")
 for(h in c(-50, 0, 50, 100)) {
   circos.lines(cell.xlim, c(h, h), col ="#00000040")
 }
}, track.height = 0.2)
circos.genomicLabels(Gene_labels, labels.column=4, side='inside', cex=0.6)
circos.clear()
dev.off()
```

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

```
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)
circos.initializeWithIdeogram(track.height = 0.05)
### Labels for inversely changing DMRs with DEG
circos.genomicDensity(DMR.PerChange, col = c("black"), track.height = 0.1, baseline="bottom", bg.border
##DEG with inverse GPI Islands Promoters
circos.genomicTrackPlotRegion(Methyl.List,
                              ylim = c(-100, 100), bg.border=NA,
                              panel.fun = function(region, value, ...) {
 col = ifelse(value[[1]] > 0, "coral2", "darkcyan")
 circos.genomicPoints(region, value, col = add_transparency(col, 0.2), cex = 0.3, pch = 16)
 cell.xlim = get.cell.meta.data("cell.xlim")
 for(h in c(-50, 0, 50, 100)) {
   circos.lines(cell.xlim, c(h, h), col ="#00000040")
```

```
 }
}, track.height = 0.2)
circos.genomicLabels(Gene_labels, labels.column=4, side='inside', cex=0.6)
```




```
circos.clear()
```

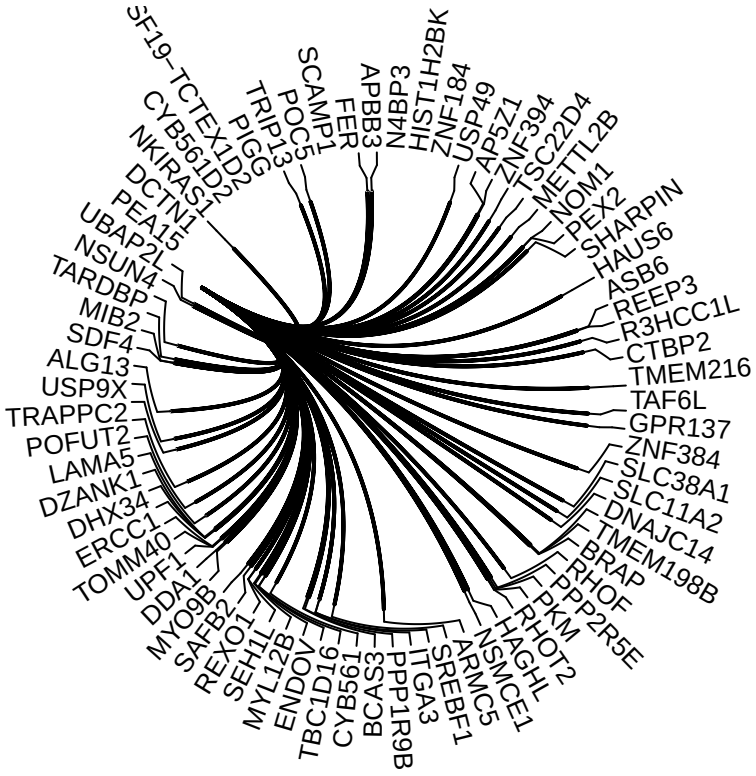
#ELK4 Motif Enrichment

```
library(dplyr)
ELK4_Anchor<-read.csv("../1_Input/ELK4/ELK4_Anchor.csv")
ELK4<-read.csv("../1_Input/ELK4/ELK4_Targets.ENCODE.csv", col.names = FALSE)
colnames(ELK4)<-"annot.symbol"
ELK4_DMR.Targets<-dplyr::inner_join(ELK4, DiffMeth_Annotated.tiles_q05)
ELK4_DMR.Targets.Promoters<-filter(ELK4_DMR.Targets,  grepl("promoters",Annotation), abs(meth.diff)>5, c
ELK4_DMR.Targets.Promoters$chr<-factor(ELK4_DMR.Targets.Promoters$chr,
                                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"))
ELK4_DMR.Targets.Promoters<-arrange(ELK4_DMR.Targets.Promoters, start)
ELK4_DMR.Targets.Promoters<-as.data.frame(ELK4_DMR.Targets.Promoters)
ELK4_DMR.Targets.Promoters<-ELK4_DMR.Targets.Promoters[!duplicated(ELK4_DMR.Targets.Promoters[,c(1)]),]
#Labels
ELK4_targets.labels<-filter(ELK4_DMR.Targets.Promoters, meth.diff>0)
```

```
ELK4_targets.labels<-ELK4_targets.labels[,c("chr", "start", "end", "annot.symbol", "meth.diff")]

#UP.only
ELK4_targets_UP<-dplyr::filter(ELK4_DMR.Targets.Promoters, meth.diff>0)
ELK4_targets_UP<-ELK4_targets_UP[,c("chr", "start", "end", "annot.symbol")]
ELK4_targets_UP<-ELK4_targets_UP %>% arrange(chr, start)
Up_anchor<-ELK4_Anchor[1:nrow(ELK4_targets_UP),]
#DOWN.only
ELK4_targets_DOWN<-dplyr::filter(ELK4_DMR.Targets.Promoters, meth.diff<0)
ELK4_targets_DOWN<-ELK4_targets_DOWN[,c("chr", "start", "end", "annot.symbol")]
ELK4_targets_DOWN<-ELK4_targets_DOWN %>% arrange(chr, start)
Down_anchor<-ELK4_Anchor[1:nrow(ELK4_targets_DOWN),]
#Anchor
ELK4_anchor<-read.csv("../1_Input/ELK4/ELK4_Anchor.csv")

#Circular Plot
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 = -190)
pdf(file=paste0("../2_Output/", ANALYSIS, "/", ANALYSIS, "_Circos_ELK4.targets.pdf"))
circos.initializeWithIdeogram(plotType = NULL)
circos.genomicLabels(ELK4_targets.labels, labels.column=4, side='outside', cex=.8)
# circos.genomicLink(ELK4_targets_DOWN, Down_anchor,
#                    col="dodgerblue3", lwd=2)
circos.genomicLink(ELK4_targets_UP, Up_anchor,
                   col="black", lwd=2)
circos.clear()
dev.off()
```

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

```
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 = -190)
circos.initializeWithIdeogram(plotType = NULL)
circos.genomicLabels(ELK4_targets.labels, labels.column=4, side='outside', cex=.8)
# circos.genomicLink(ELK4_targets_DOWN, Down_anchor,
#                    col="dodgerblue3", lwd=2)
circos.genomicLink(ELK4_targets_UP, Up_anchor,
                   col="black", lwd=2)
```



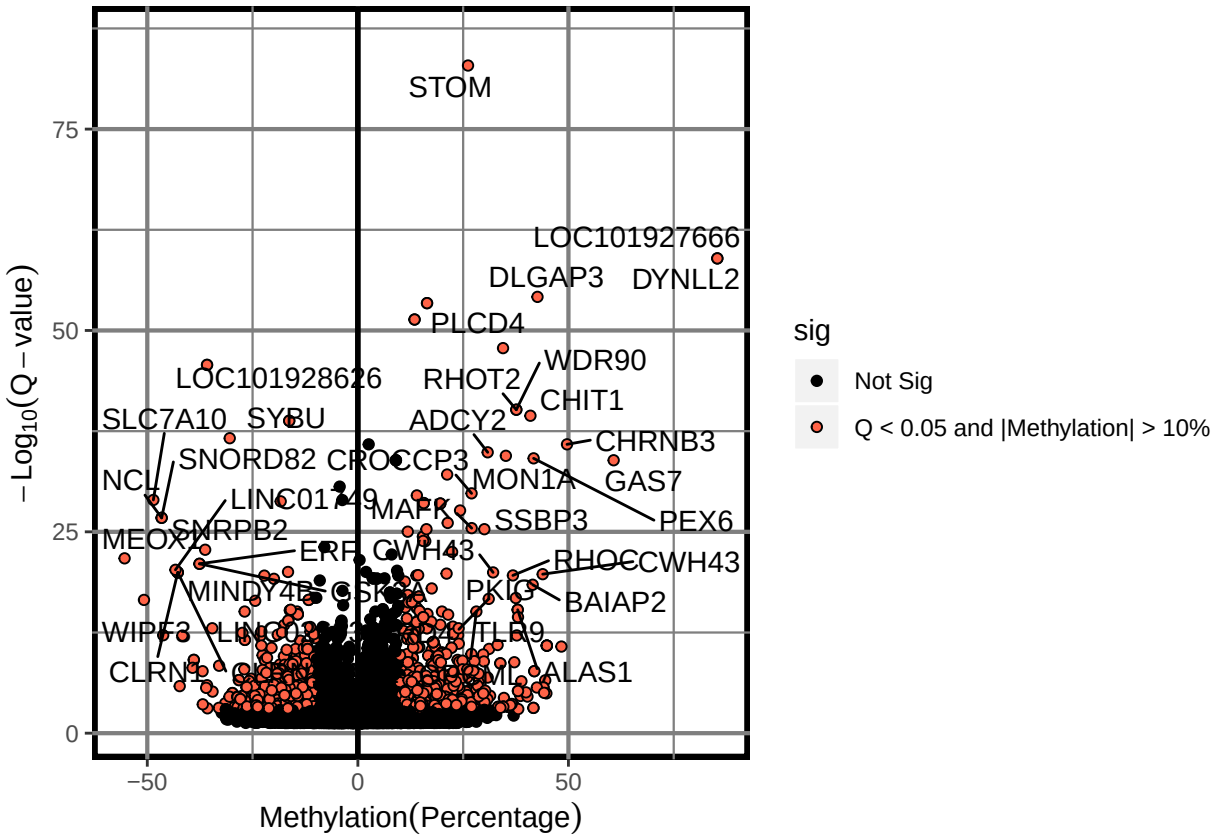

```
circos.clear()
```

##Volcano Plot

```
# Load packages
library(dplyr)
library(ggplot2)
library(ggrepel)
library(openxlsx)
library(tidyr)
# Read data from the web
results<-DiffMeth_Annotated.tiles
results<-dplyr::mutate(results, minuslogqvalue=-log(qvalue), Gene.Symbol=annot.symbol)
results$sig<-ifelse(results$qvalue<0.05 & abs(results$meth.diff)>10, "Q < 0.05 and |Methylation| > 10%"
max(results$minuslogqvalue, na.rm = TRUE)
```

```
## [1] 82.89425
```

```
#plot the ggplot
p = ggplot(results, aes(meth.diff, minuslogqvalue)) + theme(panel.background = element_rect("white", co
geom_point(aes(fill=sig), colour="black", shape=21) + labs(x=expression(Methylation(Percentage)), y=exp
scale_fill_manual(values=c("black", "tomato"))
#add a repelling effect to the text labels.
p+geom_text_repel(data=filter(results, minuslogqvalue>15 & abs(meth.diff)>25), aes(label=Gene.Symbol))
```




```
##Export Volcano Plot
pdf(file = paste0("../2_Output/", ANALYSIS, "/", ANALYSIS, "Volcano.Plot.pdf"),  width = 7.5, height = 5
p+geom_text_repel(data=filter(results, minuslogqvalue>15 & abs(meth.diff)>25), aes(label=Gene.Symbol))
dev.off()
```

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

#Supplemental Table: R Session Information

All packages and setting are acquired using the following command:

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

```
##  setting  value
##  version  R version 3.6.1 (2019-07-05)
##  os       macOS Catalina 10.15.3
##  system   x86_64, darwin15.6.0
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  ctype    en_US.UTF-8
##  tz       America/Chicago
##  date     2020-02-15
```

```
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	1.46.1	1.46.1
AnnotationHub	AnnotationHub	2.16.1	2.16.1
annotatr	annotatr	1.10.0	1.10.0
assertthat	assertthat	0.2.1	0.2.1
backports	backports	1.1.5	1.1.5
bbmle	bbmle	1.0.22	1.0.22
bdsmatrix	bdsmatrix	1.3.4	1.3-4
Biobase	Biobase	2.44.0	2.44.0
BiocFileCache	BiocFileCache	1.8.0	1.8.0
BiocGenerics	BiocGenerics	0.30.0	0.30.0
BiocManager	BiocManager	1.30.10	1.30.10
BiocParallel	BiocParallel	1.18.1	1.18.1
biomaRt	biomaRt	2.40.5	2.40.5
Biostrings	Biostrings	2.52.0	2.52.0
bit	bit	1.1.15.1	1.1-15.1
bit64	bit64	0.9.7	0.9-7
bitops	bitops	1.0.6	1.0-6
blob	blob	1.2.1	1.2.1
BSgenome	BSgenome	1.52.0	1.52.0
callr	callr	3.4.1	3.4.1
circlize	circlize	0.4.8	0.4.8
cli	cli	2.0.1	2.0.1
coda	coda	0.19.3	0.19-3
codetools	codetools	0.2.16	0.2-16
colorspace	colorspace	1.4.1	1.4-1
crayon	crayon	1.3.4	1.3.4
curl	curl	4.3	4.3
data.table	data.table	1.12.8	1.12.8
DBI	DBI	1.1.0	1.1.0
dbplyr	dbplyr	1.4.2	1.4.2
DelayedArray	DelayedArray	0.10.0	0.10.0
desc	desc	1.2.0	1.2.0
devtools	devtools	2.2.1	2.2.1
digest	digest	0.6.23	0.6.23
dplyr	dplyr	0.8.3	0.8.3
ellipsis	ellipsis	0.3.0	0.3.0
emdbook	emdbook	1.3.11	1.3.11
evaluate	evaluate	0.14	0.14
fansi	fansi	0.4.1	0.4.1

	package	ondiskversion	loadedversion
farver	farver	2.0.3	2.0.3
fastmap	fastmap	1.0.1	1.0.1
fastseg	fastseg	1.30.0	1.30.0
fs	fs	1.3.1	1.3.1
GenomeInfoDb	GenomeInfoDb	1.20.0	1.20.0
GenomeInfoDbData	GenomeInfoDbData	1.2.1	1.2.1
GenomicAlignments	GenomicAlignments	1.20.1	1.20.1
GenomicFeatures	GenomicFeatures	1.36.4	1.36.4
GenomicRanges	GenomicRanges	1.36.1	1.36.1
ggplot2	ggplot2	3.2.1	3.2.1
ggrepel	ggrepel	0.8.1	0.8.1
GlobalOptions	GlobalOptions	0.1.1	0.1.1
glue	glue	1.3.1	1.3.1
gtable	gtable	0.3.0	0.3.0
gtools	gtools	3.8.1	3.8.1
hms	hms	0.5.3	0.5.3
htmltools	htmltools	0.4.0	0.4.0
httpuv	httpuv	1.5.2	1.5.2
httr	httr	1.4.1	1.4.1
interactiveDisplayBase	interactiveDisplayBase	1.22.0	1.22.0
IRanges	IRanges	2.18.3	2.18.3
kableExtra	kableExtra	1.1.0	1.1.0
knitr	knitr	1.27	1.27
labeling	labeling	0.3	0.3
later	later	1.0.0	1.0.0
lattice	lattice	0.20.38	0.20-38
lazyeval	lazyeval	0.2.2	0.2.2
lifecycle	lifecycle	0.1.0	0.1.0
limma	limma	3.40.6	3.40.6
magrittr	magrittr	1.5	1.5
MASS	MASS	7.3.51.5	7.3-51.5
Matrix	Matrix	1.2.18	1.2-18
matrixStats	matrixStats	0.55.0	0.55.0
mclust	mclust	5.4.5	5.4.5
memoise	memoise	1.1.0	1.1.0
methylKit	methylKit	1.10.0	1.10.0
mgcv	mgcv	1.8.31	1.8-31
mime	mime	0.8	0.8
munsell	munsell	0.5.0	0.5.0
mvtnorm	mvtnorm	1.0.12	1.0-12
nlme	nlme	3.1.143	3.1-143
numDeriv	numDeriv	2016.8.1.1	2016.8-1.1
openxlsx	openxlsx	4.1.4	4.1.4
org.Hs.eg.db	org.Hs.eg.db	3.8.2	3.8.2
pheatmap	pheatmap	1.0.12	1.0.12
pillar	pillar	1.4.3	1.4.3

	package	ondiskversion	loadedversion	
pkgbuild	pkgbuild	1.0.6	1.0.6	
pkgconfig	pkgconfig	2.0.3	2.0.3	
pkgload	pkgload	1.0.2	1.0.2	
plyr	plyr	1.8.5	1.8.5	
prettyunits	prettyunits	1.1.1	1.1.1	
processx	processx	3.4.1	3.4.1	
progress	progress	1.2.2	1.2.2	
promises	promises	1.1.0	1.1.0	
ps	ps	1.3.0	1.3.0	
purrr	purrr	0.3.3	0.3.3	
qvalue	qvalue	2.16.0	2.16.0	
R.methodsS3	R.methodsS3	1.7.1	1.7.1	
R.oo	R.oo	1.23.0	1.23.0	
R.utils	R.utils	2.9.2	2.9.2	
R6	R6	2.4.1	2.4.1	
rappdirs	rappdirs	0.3.1	0.3.1	
RColorBrewer	RColorBrewer	1.1.2	1.1-2	
Rcpp	Rcpp	1.0.3	1.0.3	
RCurl	RCurl	1.98.1.1	1.98-1.1	
readr	readr	1.3.1	1.3.1	
regioneR	regioneR	1.16.5	1.16.5	
remotes	remotes	2.1.0	2.1.0	
reshape2	reshape2	1.4.3	1.4.3	
rlang	rlang	0.4.3	0.4.3	
rmarkdown	rmarkdown	2.1	2.1	
rprojroot	rprojroot	1.3.2	1.3-2	
Rsamtools	Rsamtools	2.0.3	2.0.3	
RSQLite	RSQLite	2.2.0	2.2.0	
rstudioapi	rstudioapi	0.10	0.10	
rtracklayer	rtracklayer	1.44.4	1.44.4	
rvest	rvest	0.3.5	0.3.5	
S4Vectors	S4Vectors	0.22.1	0.22.1	
scales	scales	1.1.0	1.1.0	
sessioninfo	sessioninfo	1.1.1	1.1.1	
shape	shape	1.4.4	1.4.4	
shiny	shiny	1.4.0	1.4.0	
stringi	stringi	1.4.5	1.4.5	
stringr	stringr	1.4.0	1.4.0	
SummarizedExperiment	SummarizedExperiment	1.14.1	1.14.1	
testthat	testthat	2.3.1	2.3.1	
tibble	tibble	2.1.3	2.1.3	
tidyr	tidyr	1.0.2	1.0.2	
tidyselect	tidyselect	0.2.5	0.2.5	
TxDb.Hsapiens.UCSC.hg38.knownGene	TxDb.Hsapiens.UCSC.hg38.knownGene	3.4.6	3.4.6	/Libr
usethis	usethis	1.5.1	1.5.1	
vctrs	vctrs	0.2.2	0.2.2	

	package	ondiskversion	loadedversion
viridisLite	viridisLite	0.3.0	0.3.0
webshot	webshot	0.5.2	0.5.2
withr	withr	2.1.2	2.1.2
xfun	xfun	0.12	0.12
XML	XML	3.99.0.3	3.99-0.3
xml2	xml2	1.2.2	1.2.2
xtable	xtable	1.8.4	1.8-4
XVector	XVector	0.24.0	0.24.0
yaml	yaml	2.2.0	2.2.0
zip	zip	2.0.4	2.0.4
zlibbioc	zlibbioc	1.30.0	1.30.0