

# Fig7V4

zacharyLewis

2024-08-16

```
if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager", type = "source")

## Warning: package 'BiocManager' was built under R version 4.3.3

## Bioconductor version '3.17' is out-of-date; the current release version '3.19'
##   is available with R version '4.4'; see https://bioconductor.org/install

BiocManager::install("csaw") #INSTALL from source (I had to download a gfortran installer from CRAN (ht

## Bioconductor version 3.17 (BiocManager 1.30.24), R 4.3.1 (2023-06-16)

## Warning: package(s) not installed when version(s) same as or greater than current; use
##   'force = TRUE' to re-install: 'csaw'

## Old packages: 'httr2', 'PieGlyph', 'pkgdown', 'RcppArmadillo', 'RcppEigen',
##   'reticulate', 'waldo'

if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install("DiffBind")

## Bioconductor version 3.17 (BiocManager 1.30.24), R 4.3.1 (2023-06-16)

## Warning: package(s) not installed when version(s) same as or greater than current; use
##   'force = TRUE' to re-install: 'DiffBind'

## Old packages: 'httr2', 'PieGlyph', 'pkgdown', 'RcppArmadillo', 'RcppEigen',
##   'reticulate', 'waldo'

if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install("profileplyr")

## Bioconductor version 3.17 (BiocManager 1.30.24), R 4.3.1 (2023-06-16)
```

```

## Warning: package(s) not installed when version(s) same as or greater than current; use
##   'force = TRUE' to re-install: 'profileplyr'

## Old packages: 'httr2', 'PieGlyph', 'pkgdown', 'RcppArmadillo', 'RcppEigen',
##   'reticulate', 'waldo'

#edgeR
if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install("edgeR")

## Bioconductor version 3.17 (BiocManager 1.30.24), R 4.3.1 (2023-06-16)

## Warning: package(s) not installed when version(s) same as or greater than current; use
##   'force = TRUE' to re-install: 'edgeR'

## Old packages: 'httr2', 'PieGlyph', 'pkgdown', 'RcppArmadillo', 'RcppEigen',
##   'reticulate', 'waldo'

if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install("ComplexHeatmap")

## Bioconductor version 3.17 (BiocManager 1.30.24), R 4.3.1 (2023-06-16)

## Warning: package(s) not installed when version(s) same as or greater than current; use
##   'force = TRUE' to re-install: 'ComplexHeatmap'

## Old packages: 'httr2', 'PieGlyph', 'pkgdown', 'RcppArmadillo', 'RcppEigen',
##   'reticulate', 'waldo'

#https://bioconductor.org/packages/release/bioc/manuals/csaw/man/csaw.pdf

#MACS: Call peaks on WT K9me3 and WT K27me3
library(csaw)

## Loading required package: GenomicRanges

## Loading required package: stats4

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:stats':
## 
##     IQR, mad, sd, var, xtabs

```

```

## The following objects are masked from 'package:base':
##
##      anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##      colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##      get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##      match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##      Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##      table, tapply, union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:utils':
##
##      findMatches

## The following objects are masked from 'package:base':
##
##      expand.grid, I, unname

## Loading required package: IRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

## Warning: package 'matrixStats' was built under R version 4.3.2

##
## Attaching package: 'MatrixGenerics'

## The following objects are masked from 'package:matrixStats':
##
##      colAlls, colAnyNAs, colAnyNs, colAvgsPerRowSet, colCollapse,
##      colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##      colDiffss, colIQRDiffss, colIQRs, colLogSumExps, colMadDiffss,
##      colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##      colProds, colQuantiles, colRanges, colRanks, colSdDiffss, colSds,
##      colSums2, colTabulates, colVarDiffss, colVars, colWeightedMads,
##      colWeightedMeans, colWeightedMedians, colWeightedSds,
##      colWeightedVars, rowAlls, rowAnyNAs, rowAnyNs, rowAvgsPerColSet,
##      rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##      rowCumsums, rowDiffss, rowIQRDiffss, rowIQRs, rowLogSumExps,
##      rowMadDiffss, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##      rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##      rowSdDiffss, rowSds, rowSums2, rowTabulates, rowVarDiffss, rowVars,
##      rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##      rowWeightedSds, rowWeightedVars

```

```

## Loading required package: Biobase

## Welcome to Bioconductor
##
##      Vignettes contain introductory material; view with
##      'browseVignettes()'. To cite Bioconductor, see
##      'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':
## 
##     rowMedians

## The following objects are masked from 'package:matrixStats':
## 
##     anyMissing, rowMedians

getwd()

## [1] "/Users/zacharylewis/Dropbox/DropBOX Documents/2. Zack Papers/2024_HDA1_paper/DataForFigures/Fig

#read in sample sheet
samples <- read.csv("./Figure7_SampleSheet.csv")

samples$bamReads <- paste("./bamFiles/",samples$bamReads, sep="")

K27samples <- samples[samples$Antibody == "H3K27me3", ]
K9samples <- samples[samples$Antibody == "H3K9me3", ]
K36samples <- samples[samples$Antibody == "H3K36me3",]
Kacsamples <- samples[samples$Antibody == "Kac",]

##Step 1: create counts for all bins excluding the mat locus, which may be different in different strains
#####remove mat locus
matLocus <- GRanges("CM002236.1", IRanges(1856288, 1862459)) # mat

##set parameters for windowCounts
param <- readParam(discard = matLocus, pe="both", max.frag=1500, minq=20 )

#read counts from bam files
data_csaw <- windowCounts(samples$bamReads, spacing=300, width=300, shift=0,
filter=0, bin=TRUE, param=param)

#data_K27 <- windowCounts(K27samples$bamReads, spacing=1000, width=1000, shift=0,
#filter=0, bin=TRUE, param=param)

#####QC check: look at fragment sizes of PE data to get max.frag parameter used above.
# out <- getPESizes(samples$bamReads)
# frag.sizes <- out$sizes[out$sizes<= 1500]
# hist(frag.sizes, breaks=50, xlab="Fragment sizes (bp)", ylab="Frequency", main="", col="grey80")

```

```

# #inspect the counts
# head(assay(data_csa))
# #view the window size locations
# head(rowRanges(data_csa))
# #view the sample names
# head(colData(data_csa))
# #preview totals
# data_csa$totals

##identify high abundance windows and normalize to correct for chip efficiency
# K27.bin <- windowCounts(K27samples$bamReads, bin=TRUE, width=10000, param=param)
# keep.K27 <- filterWindowsGlobal(data_K27, K27.bin)$filter > log2(3)
# K27.filtered <- data_K27[keep.K27,]
#
# K27.filtered <- normFactors(K27.filtered)
# K27.eff <- K27.filtered$norm.factors
# K27.eff
### don't like this one. try selecting only K27/K9 peaks

K27peaks <- read.table(file="./141-1_ChIP_WT_H3K27me3__peaks.broadPeak")
K9peaks <- read.table(file="./141-115_ChIP_WT_H3K9me3__peaks.broadPeak")
Regions <- rbind(K27peaks, K9peaks)

Regions.gr <- GRanges(Regions$V1, IRanges(Regions$V2, Regions$V3))

K27_Peaks.gr <- GRanges(K27peaks$V1, IRanges(K27peaks$V2, K27peaks$V3))
K9_Peaks.gr <- GRanges(K9peaks$V1, IRanges(K9peaks$V2, K9peaks$V3))

suppressWarnings(keep <- overlapsAny(rowRanges(data_csa), Regions.gr, type = "within"))
summary(keep)

##      Mode    FALSE     TRUE
## logical 106016   31003

#filtered data keeping all K9 and K27 regions
filtered.data <- data_csa[keep,]

#make a granges object and a dataframe object for K9 domains
suppressWarnings(keepK9 <- overlapsAny(rowRanges(data_csa), K9_Peaks.gr, type="within"))
summary(keepK9)

##      Mode    FALSE     TRUE
## logical 115812   21207

filtered.data.K9 <- data_csa[keepK9,]
K9.Windows.df <- as.data.frame(granges(filtered.data.K9))

#make a granges object and a dataframe object for K27 domains

```

```

suppressWarnings(keepK27 <- overlapsAny(rowRanges(data_csa), K27_Peaks.gr, type="within"))
summary(keepK27)

##      Mode    FALSE     TRUE
## logical 126410   10609

filtered.data.K27 <- data_csa[keepK27,]
K27.Windows.df <- as.data.frame(granges(filtered.data.K27))

#remove regions present in both K9 and K9 peaks e.g. subtelomeres
k9andk27_regions <- filtered.data.K9[filtered.data.K9 %over% filtered.data.K27,]
minusShared <- filtered.data[!filtered.data %over% k9andk27_regions]
suppressWarnings(keep <- overlapsAny(rowRanges(data_csa), minusShared, type = "within"))
summary(keep)

##      Mode    FALSE     TRUE
## logical 106829   30190

filtered.data <- data_csa[keep,]

##Try setting up edgeR for each modification. This should better estimate dispersion

##### Analyze differential binding events

library(edgeR)

## Loading required package: limma

##
## Attaching package: 'limma'

## The following object is masked from 'package:BiocGenerics':
##       plotMA

#define groups for analysis (WT versus mutant timepoints)

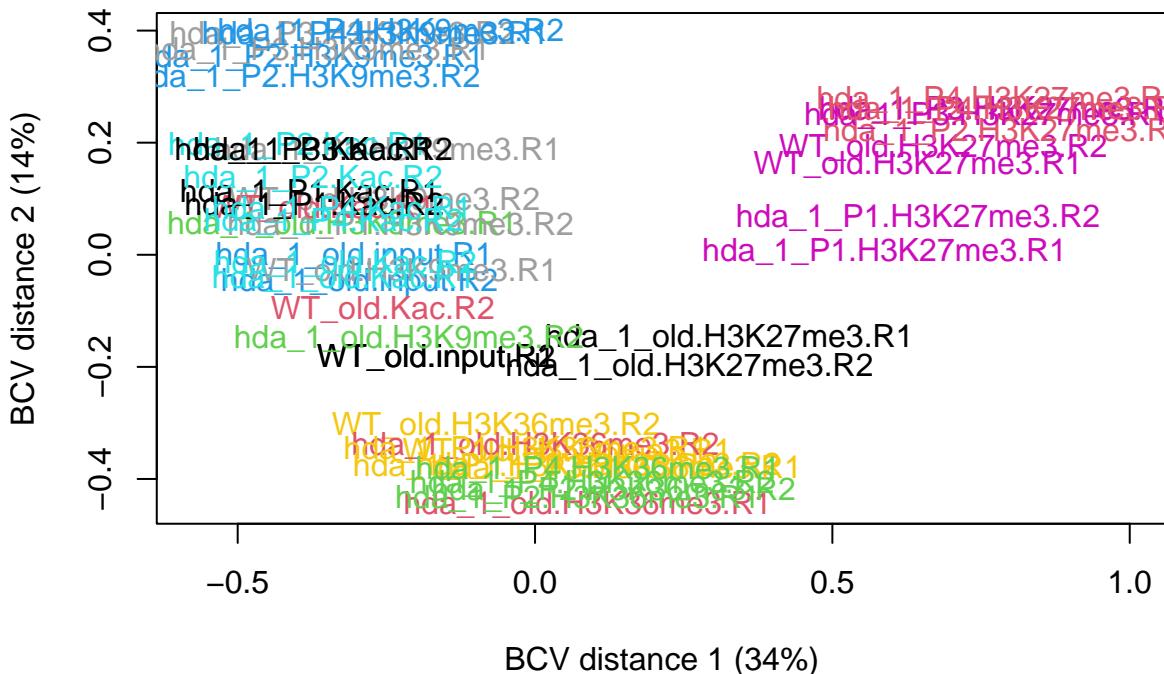
grouping <- factor(paste(samples$Strain, samples$Antibody, sep="."))
#this creates an object with a counts DF and a samples DF

#create a DGEListK9regions object
DGEList <- asDGEList(filtered.data, group=factor(grouping))
#replace generic "Sample1, Sample2" names with actual names of samples
colnames(DGEList$counts) <- paste(paste(samples$Strain,samples$Antibody, "R", sep="."), samples$Replica)
rownames(DGEList$samples) <- paste(paste(samples$Strain,samples$Antibody, "R", sep="."), samples$Replica)

#examine data
plotMDS(DGEList, method="bca", col=as.numeric(DGEList$samples$group))

```

```
## Note: the bcv method is now scheduled to be removed in a future release of edgeR.
```



```
#plot shows replicates behave well.

#d1 <- estimateCommonDisp(DGElist, verbose=T)

#design a model matrix to compare groups
design.mat <- model.matrix(~0 + DGElist$samples$group)
#replace wonky column names with readable ones
colnames(design.mat) <- levels(DGElist$samples$group)

#####
####
####
#      Estimate Dispersion
####
#####
y <- estimateDisp(DGElist, design.mat)
summary(y$trended.dispersion)
```

```
##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## 0.07292 0.07500 0.07753 0.07777 0.07987 0.13239
```

```
fit <- glmQLFit(y, design.mat, robust=TRUE)
summary(fit$var.post)
```

```
##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## 0.781   1.025   1.105   1.124   1.199   2.934
```

```
head(y$counts)
```

```
##    WT_old.input.R1 WT_old.input.R2 hda_1_old.input.R1 hda_1_old.input.R2
## 1          103          103         33          48
## 2          99           82         18          45
## 3          81           74         27          37
## 4         111          111         21          53
## 5          78           84         39          59
## 6         114          101         28          66
##    WT_old.H3K27me3.R1 WT_old.H3K27me3.R2 hda_1_old.H3K27me3.R1
## 1          818         1139         73
## 2         1226          990         43
## 3          948          780         52
## 4          600          583         61
## 5          577          469        103
## 6          325          354        140
##    hda_1_old.H3K27me3.R2 hda_1_P1.H3K27me3.R1 hda_1_P1.H3K27me3.R2
## 1          38          255        387
## 2          29          414        481
## 3          30          373        374
## 4          42          205        216
## 5          40          261        288
## 6          50          192        217
##    hda_1_P2.H3K27me3.R1 hda_1_P2.H3K27me3.R2 hda_1_P3.H3K27me3.R1
## 1         2871          861        768
## 2         3796         1449        924
## 3         3074         1095        717
## 4         1857          703        468
## 5         2186          780        481
## 6         1580          675        418
##    hda_1_P3.H3K27me3.R2 hda_1_P4.H3K27me3.R1 hda_1_P4.H3K27me3.R2
## 1         744          1409        721
## 2         962          1547        699
## 3         760          1214        552
## 4         420          732        391
## 5         449          855        355
## 6         375          589        329
##    WT_old.H3K9me3.R1 WT_old.H3K9me3.R2 hda_1_old.H3K9me3.R1 hda_1_old.H3K9me3.R2
## 1          49          345         85          88
## 2          8           219         44          93
## 3          20           129         29          58
## 4          17           107         54          55
## 5          19           125         54          50
## 6          38            95         59          40
##    hda_1_P1.H3K9me3.R1 hda_1_P1.H3K9me3.R2 hda_1_P2.H3K9me3.R1
## 1         164          185        268
## 2          55           87        122
## 3          39           69        86
## 4          27           46        61
## 5          29           38        60
## 6          26           56        52
##    hda_1_P2.H3K9me3.R2 hda_1_P3.H3K9me3.R1 hda_1_P3.H3K9me3.R2
## 1         257           47          68
```

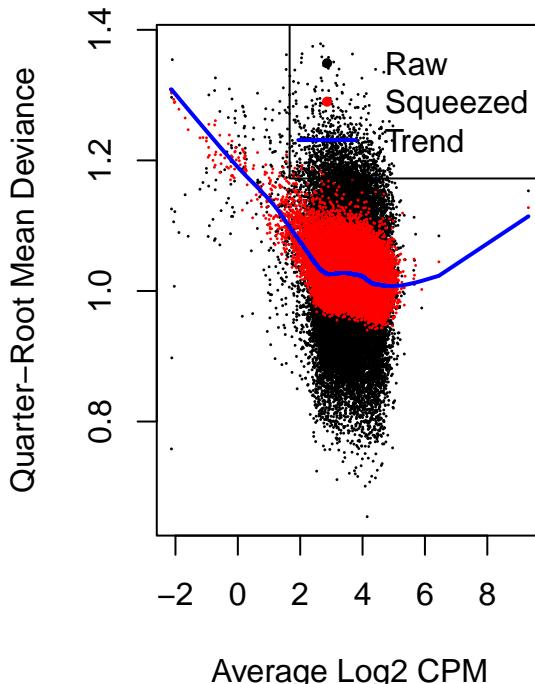
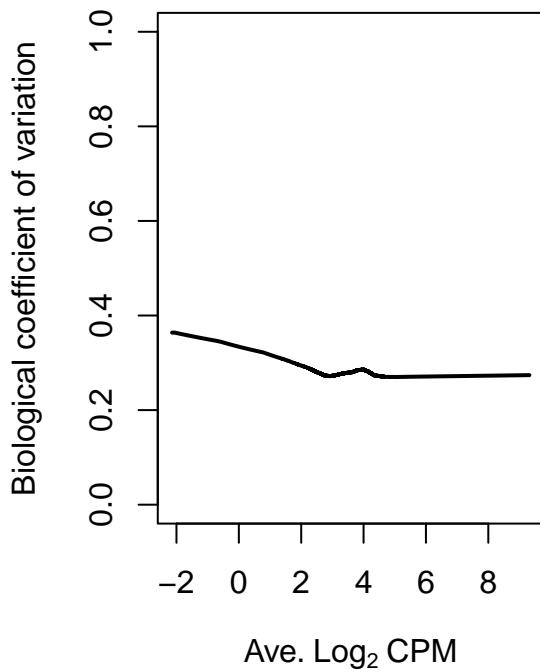
## 2	154	40	38
## 3	74	9	11
## 4	53	9	7
## 5	66	7	13
## 6	61	8	16
## hda_1_P4.H3K9me3.R1	hda_1_P4.H3K9me3.R2	WT_old.H3K36me3.R1	WT_old.H3K36me3.R2
## 1	78	60	48
## 2	18	19	36
## 3	10	8	19
## 4	10	6	17
## 5	18	9	19
## 6	25	3	20
## hda_1_old.H3K36me3.R1	hda_1_old.H3K36me3.R2	hda_1_P1.H3K36me3.R1	
## 1	78	34	57
## 2	69	46	26
## 3	59	20	38
## 4	45	9	15
## 5	28	10	15
## 6	62	10	17
## hda_1_P1.H3K36me3.R2	hda_1_P2.H3K36me3.R1	hda_1_P2.H3K36me3.R2	
## 1	40	34	65
## 2	41	41	69
## 3	17	32	36
## 4	17	18	22
## 5	20	19	23
## 6	23	20	18
## hda_1_P3.H3K36me3.R1	hda_1_P3.H3K36me3.R2	hda_1_P4.H3K36me3.R1	
## 1	36	8	53
## 2	49	9	28
## 3	22	8	18
## 4	11	5	7
## 5	7	4	4
## 6	14	0	10
## hda_1_P4.H3K36me3.R2	WT_old.Kac.R1	WT_old.Kac.R2	hda_1_old.Kac.R1
## 1	89	29	90
## 2	68	77	123
## 3	51	88	105
## 4	20	56	80
## 5	21	97	67
## 6	20	99	55
## hda_1_old.Kac.R2	hda_1_P1.Kac.R1	hda_1_P1.Kac.R2	hda_1_P2.Kac.R1
## 1	48	16	9
## 2	120	39	45
## 3	79	35	44
## 4	67	43	31
## 5	101	73	49
## 6	75	30	25
## hda_1_P2.Kac.R2	hda_1_P3.Kac.R1	hda_1_P3.Kac.R2	hda_1_P4.Kac.R1
## 1	76	21	192
## 2	88	41	186
## 3	102	18	182
## 4	90	31	174
## 5	107	29	207
## 6	96	30	132
			14

```

##     hda_1_P4.Kac.R2
## 1          50
## 2          70
## 3         99
## 4         47
## 5         49
## 6         43

#plot fit
par(mfrow=c(1,2))
o <- order(y$AveLogCPM)
plot(y$AveLogCPM[o], sqrt(y$trended.dispersion[o]), type="l", lwd=2,
      ylim=c(0, 1), xlab=expression("Ave."~Log[2]~"CPM"),
      ylab=("Biological coefficient of variation"))
plotQLDisp(fit)

```



To do: #1 Do individual comparisons to WT #2 only plot FC data if it is different than WT. #3 insignificant data should be transformed to log2 0 for heatmap.

```

##P1 versus WT
FC_P1vWT.Kac.contrast <- makeContrasts(Kac_P1vWT=hda_1_P1.Kac-WT_old.Kac, levels=design.mat)
FC_P1vWT.Kac.res <- glmQLFTest(fit, contrast=FC_P1vWT.Kac.contrast)
FC_P1vWT.Kac.merged <- mergeResults(filtered.data, FC_P1vWT.Kac.res$table, tol=300,
                                         merge.args=list(max.width=1000))
P1vWT.Kac.tabular <- FC_P1vWT.Kac.merged$combined
P1vWT.Kac.out.ranges <- FC_P1vWT.Kac.merged$regions
mcols(P1vWT.Kac.out.ranges) <- DataFrame(P1vWT.Kac.tabular)

##P2 versus WT
FC_P2vWT.Kac.contrast <- makeContrasts(Kac_P2vWT=hda_1_P2.Kac-WT_old.Kac, levels=design.mat)
FC_P2vWT.Kac.res <- glmQLFTest(fit, contrast=FC_P2vWT.Kac.contrast)
FC_P2vWT.Kac.merged <- mergeResults(filtered.data, FC_P2vWT.Kac.res$table, tol=300,
                                         merge.args=list(max.width=1000))

```

```

        merge.args=list(max.width=1000))
P2vWT.Kac.tabular <- FC_P2vWT.Kac.merged$combined
P2vWT.Kac.out.ranges <- FC_P2vWT.Kac.merged$regions
mcols(P2vWT.Kac.out.ranges) <- DataFrame(P2vWT.Kac.tabular)

##P3 versus WT
FC_P3vWT.Kac.contrast <- makeContrasts(Kac_P3vWT=hda_1_P3.Kac-WT_old.Kac,levels=design.mat)
FC_P3vWT.Kac.res <- glmQLFTest(fit, contrast=FC_P3vWT.Kac.contrast)
FC_P3vWT.Kac.merged <- mergeResults(filtered.data, FC_P3vWT.Kac.res$table, tol=300,
                                       merge.args=list(max.width=1000))
P3vWT.Kac.tabular <- FC_P3vWT.Kac.merged$combined
P3vWT.Kac.out.ranges <- FC_P3vWT.Kac.merged$regions
mcols(P3vWT.Kac.out.ranges) <- DataFrame(P3vWT.Kac.tabular)

##HDA1_old versus WT
FC_HDA1_oldvWT.Kac.contrast <- makeContrasts(Kac_HDA1_oldvWT=hda_1_old.Kac-WT_old.Kac,levels=design.mat)
FC_HDA1_oldvWT.Kac.res <- glmQLFTest(fit, contrast=FC_HDA1_oldvWT.Kac.contrast)
FC_HDA1_oldvWT.Kac.merged <- mergeResults(filtered.data, FC_HDA1_oldvWT.Kac.res$table, tol=300,
                                             merge.args=list(max.width=1000))
HDA1_oldvWT.Kac.tabular <- FC_HDA1_oldvWT.Kac.merged$combined
HDA1_oldvWT.Kac.out.ranges <- FC_HDA1_oldvWT.Kac.merged$regions
mcols(HDA1_oldvWT.Kac.out.ranges) <- DataFrame(HDA1_oldvWT.Kac.tabular)

Kac.chip.V.WT.res.df <- data.frame(seqnames=seqnames(P1vWT.Kac.out.ranges),
                                      start=start(P1vWT.Kac.out.ranges),
                                      end=end(P1vWT.Kac.out.ranges),
                                      names=c(rep(".", length(P1vWT.Kac.out.ranges))),
                                      scores=c(rep(".", length(P1vWT.Kac.out.ranges))),
                                      strand=strand(P1vWT.Kac.out.ranges),
                                      logFC_P1vWT_Kac=P1vWT.Kac.out.ranges$rep.logFC,
                                      PValue_P1vWT_Kac=P1vWT.Kac.out.ranges$PValue,
                                      FDR_P1vWT_Kac=P1vWT.Kac.out.ranges$FDR,
                                      logFC_P2vWT_Kac=P2vWT.Kac.out.ranges$rep.logFC,
                                      PValue_P2vWT_Kac=P2vWT.Kac.out.ranges$PValue,
                                      FDR_P2vWT_Kac=P2vWT.Kac.out.ranges$FDR,
                                      logFC_P3vWT_Kac=P3vWT.Kac.out.ranges$rep.logFC,
                                      PValue_P3vWT_Kac=P3vWT.Kac.out.ranges$PValue,
                                      FDR_P3vWT_Kac=P3vWT.Kac.out.ranges$FDR,
                                      logFC_HDA1_oldvWT_Kac=HDA1_oldvWT.Kac.out.ranges$rep.logFC,
                                      PValue_HDA1_oldvWT_Kac=HDA1_oldvWT.Kac.out.ranges$PValue,
                                      FDR_HDA1_oldvWT_Kac=HDA1_oldvWT.Kac.out.ranges$FDR)

Kac.chip.V.WT.res.df_SigOnly <- within(Kac.chip.V.WT.res.df, logFC_P1vWT_Kac[FDR_P1vWT_Kac > 0.05] <- 1)
Kac.chip.V.WT.res.df_SigOnly <- within(Kac.chip.V.WT.res.df_SigOnly, logFC_P2vWT_Kac[FDR_P2vWT_Kac > 0.05] <- 1)
Kac.chip.V.WT.res.df_SigOnly <- within(Kac.chip.V.WT.res.df_SigOnly, logFC_P3vWT_Kac[FDR_P3vWT_Kac > 0.05] <- 1)
Kac.chip.V.WT.res.df_SigOnly <- within(Kac.chip.V.WT.res.df_SigOnly, logFC_HDA1_oldvWT_Kac[FDR_HDA1_oldvWT_Kac > 0.05] <- 1)

KacVwt_Heatmap_df <- dplyr::select(Kac.chip.V.WT.res.df_SigOnly, 7, 10, 13, 16)
rownames(KacVwt_Heatmap_df) <- paste(Kac.chip.V.WT.res.df_SigOnly$seqnames, Kac.chip.res.df_SigOnly$start, Kac.chip.res.df_SigOnly$end, sep=":")

```

```

#df1.K9peaksOnly <- df1 %>% dplyr::inner_join(K9.Windows.df, by=c("seqnames", "start", "end"))
#df1.K27peaksOnly <- df1 %>% dplyr::inner_join(K27.Windows.df, by=c("seqnames", "start", "end"))

library(pheatmap)
library(ComplexHeatmap)

## Loading required package: grid

## =====
## ComplexHeatmap version 2.16.0
## Bioconductor page: http://bioconductor.org/packages/ComplexHeatmap/
## Github page: https://github.com/jokergoo/ComplexHeatmap
## Documentation: http://jokergoo.github.io/ComplexHeatmap-reference
##
## If you use it in published research, please cite either one:
## - Gu, Z. Complex Heatmap Visualization. iMeta 2022.
## - Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional
##   genomic data. Bioinformatics 2016.
##
## 
## The new InteractiveComplexHeatmap package can directly export static
## complex heatmaps into an interactive Shiny app with zero effort. Have a try!
##
## This message can be suppressed by:
## suppressPackageStartupMessages(library(ComplexHeatmap))
## =====
## ! pheatmap() has been masked by ComplexHeatmap::pheatmap(). Most of the arguments
##   in the original pheatmap() are identically supported in the new function. You
##   can still use the original function by explicitly calling pheatmap::pheatmap().

##
## Attaching package: 'ComplexHeatmap'

## The following object is masked from 'package:pheatmap':
## 
##     pheatmap

library(RColorBrewer)
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.3.2

#remove outliers/set heatmap scale
K9_Heatmap_mat <- as.matrix(K9_Heatmap_df)
quantile(K9_Heatmap_mat, c(.01, .50, .75, .95, .98, .995))

##          1%        50%       75%       95%       98%      99.5%
## -3.142955  1.592224  2.312543  3.097624  3.421188  3.990779

```

```

max <- quantile(K9_Heatmap_mat, 0.99) # identify the 98th percentile value and assign this to max
K9_Heatmap_mat[K9_Heatmap_mat>max] = max      #replace any value larger than the 98th percentile value wi

K27_Heatmap_mat <- as.matrix(K27_Heatmap_df)
quantile(K27_Heatmap_mat, c(.01, .50, .75, .95, .98 , .995))

##           1%        50%        75%        95%        98%       99.5%
## -3.191381  0.100000  2.056443  4.113028  4.589671  5.018174

max <- quantile(K27_Heatmap_mat, 0.99) # identify the 98th percentile value and assign this to max
K27_Heatmap_mat[K27_Heatmap_mat>max] = max      #replace any value larger than the 98th percentile value wi

#MERGE K9 AND K27 DATA
K27datK9dat <- cbind(K27_Heatmap_mat, K9_Heatmap_mat)

#
#set up color palette
# col <- colorRampPalette(c("white", "blue"))(n = 299)
# # (optional) defines the color breaks manually for a "skewed" color transition
# breaks = c(-2, seq(0.11, 3.5,length=100))
#
#
# order_heatmap<- pheatmap(K27datK9dat, breaks = breaks, color = col , cellwidth = NA, scale="none", ce
#
#
# #colorRampPalette(brewer.pal(n = 7, name ="Blues"))(1000)
#
#
# #to plot with ggplot, you need to extract [[4]] from the heatmap object
# ggsave(heatmap_plot, filename = "./order.png", height = 5 , width = 5, limitsize = FALSE)

####get the row order of another pheatmap and set the first to the second/https://www.biostars.org/p/214
####hm_2$tree$row$order <- hm_1$tree$row$order

####test complex heatmap
library(circlize)

## Warning: package 'circlize' was built under R version 4.3.2

## =====
## circlize version 0.4.16
## CRAN page: https://cran.r-project.org/package=circlize
## Github page: https://github.com/jokergoo/circlize
## Documentation: https://jokergoo.github.io/circlize_book/book/
##
## If you use it in published research, please cite:
## Gu, Z. circlize implements and enhances circular visualization
## in R. Bioinformatics 2014.
##
## This message can be suppressed by:
## suppressPackageStartupMessages(library(circlize))
## =====

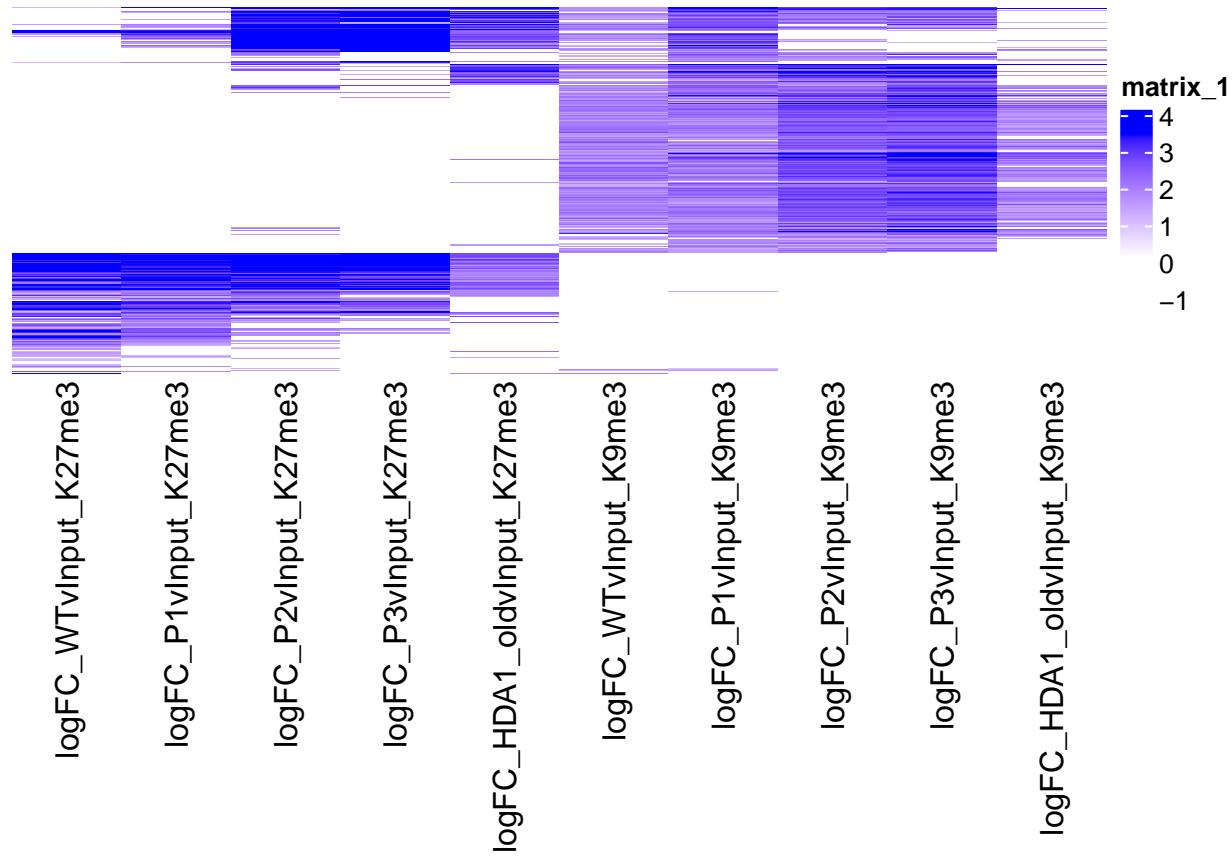
```

```
col_fun<- colorRamp2(c(-1, .11, 3.5), c("white", "white", "blue"))
col_fun(seq(-1, 3.5, length = 20))
```

```
## [1] "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FCFAFFF"
## [7] "#F3E9FFFF" "#E9D9FFFF" "#DFC9FFFF" "#D4B9FFFF" "#C9A9FFFF" "#BD99FFFF"
## [13] "#B189FFFF" "#A479FFFF" "#9569FFFF" "#8558FFFF" "#7448FFFF" "#5E36FFFF"
## [19] "#4222FFFF" "#0000FFFF"
```

*#default is euclidean and complete*

```
heatmap_ORDER <- Heatmap(K27datK9dat, col = col_fun, cluster_columns = F, show_row_names = FALSE, show_col_names = FALSE)
ht <- draw(heatmap_ORDER )
```



```
#try removing windows that are present in both WT K9 and K27 peaks # Make df with K27me3 data #
#cluster based on K9 and K27 data # dump cluster order # make
```

Description of Heatmap - reads were counted in 1000 bp bins overlapping K9 or K27 peaks. Windows overlapping both K27 and K9 peaks were removed. windows with LogFC < 2 or FDR > 0.5 are shown in white.

```
#remove outliers/set heatmap scale for each mod
```

#K27

```
K27_Heatmap_mat <- as.matrix(K27_Heatmap_df)
quantile(K27_Heatmap_mat, c(.01, .50, .75, .95, .98, .995))
```

```

##           1%          50%          75%          95%          98%         99.5%
## -3.191381  0.100000  2.056443  4.113028  4.589671  5.018174

max <- quantile(K27_Heatmap_mat, 0.99) # identify the 98th percentile value and assign this to max
K27_Heatmap_mat[K27_Heatmap_mat>max] = max      #replace any value larger than the 98th percentile value with max

#K9
K9_Heatmap_mat <- as.matrix(K9_Heatmap_df)
quantile(K9_Heatmap_mat, c(.01, .50, .75, .95, .98, .995))

##           1%          50%          75%          95%          98%         99.5%
## -3.142955  1.592224  2.312543  3.097624  3.421188  3.990779

max <- quantile(K9_Heatmap_mat, 0.99) # identify the 98th percentile value and assign this to max
K9_Heatmap_mat[K9_Heatmap_mat>max] = max      #replace any value larger than the 98th percentile value with max

#K36
K36_Heatmap_mat <- as.matrix(K36_Heatmap_df)
quantile(K36_Heatmap_mat, c(.01, .50, .75, .95, .98, .995))

##           1%          50%          75%          95%          98%         99.5%
## -3.532747  0.100000  0.100000  2.025983  2.336970  2.728655

max <- quantile(K36_Heatmap_mat, 0.99) # identify the 98th percentile value and assign this to max
K36_Heatmap_mat[K36_Heatmap_mat>max] = max      #replace any value larger than the 98th percentile value with max

#Kac v input
Kac_Heatmap_mat <- as.matrix(Kac_Heatmap_df)
quantile(Kac_Heatmap_mat, c(.01, .50, .75, .95, .98, .995))

##           1%          50%          75%          95%          98%         99.5%
## -3.553221 -1.000000 -1.000000  1.497522  1.863134  2.373427

max <- quantile(Kac_Heatmap_mat, 0.99) # identify the 98th percentile value and assign this to max
Kac_Heatmap_mat[Kac_Heatmap_mat>max] = max      #replace any value larger than the 98th percentile value with max

#Kac v WT
KacVwt_Heatmap_mat <- as.matrix(KacVwt_Heatmap_df)
quantile(KacVwt_Heatmap_mat, c(.01, .50, .75, .95, .98, .995))

##           1%          50%          75%          95%          98%         99.5%
## -0.100000 -0.100000  2.944164  4.260338  4.716772  5.422414

max <- quantile(KacVwt_Heatmap_mat, 0.99) # identify the 98th percentile value and assign this to max
KacVwt_Heatmap_mat[KacVwt_Heatmap_mat>max] = max      #replace any value larger than the 98th percentile value with max

#K27
library(circlize)
col_fun<- colorRamp2(c(-1, .2, quantile(K27_Heatmap_mat, 0.99)), c("white", "white", "darkgreen"))
col_fun(seq(-1, quantile(K27_Heatmap_mat, 0.99), length = 20))

```

```

## [1] "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FEFEDFF" "#EEF3ECFF"
## [7] "#DFE9DAFF" "#CFDEC9FF" "#COD4B8FF" "#B1C9A8FF" "#A1BF97FF" "#92B587FF"
## [13] "#83AA77FF" "#74A067FF" "#659657FF" "#568C48FF" "#468238FF" "#347828FF"
## [19] "#216E17FF" "#006400FF"

K27_hm <- Heatmap(K27_Heatmap_mat, col = col_fun, cluster_columns = F, show_row_names = FALSE, show_col

## Warning: The heatmap has not been initialized. You might have different results
## if you repeatedly execute this function, e.g. when row_km/column_km was
## set. It is more suggested to do as 'ht = draw(ht); row_order(ht)'.

#K9
col_fun<- colorRamp2(c(-1, .2, quantile(K9_Heatmap_mat, 0.99)), c("white", "white", "blue"))
col_fun(seq(-1, quantile(K9_Heatmap_mat, 0.99), length = 20))

## [1] "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FEFDFFFF"
## [7] "#F4ECFFFF" "#EBDCFFFF" "#EOCBFFFF" "#D6BBFFFF" "#CAABFFFF" "#BF9AFFFF"
## [13] "#B28AFFFF" "#A57AFFFF" "#966AFFFF" "#8659FFFF" "#7448FFFF" "#5F36FFFF"
## [19] "#4322FFFF" "#0000FFFF"

K9_hm <- Heatmap(K9_Heatmap_mat, col = col_fun, cluster_columns = F, show_row_names = FALSE, show_col

## Warning: The heatmap has not been initialized. You might have different results
## if you repeatedly execute this function, e.g. when row_km/column_km was
## set. It is more suggested to do as 'ht = draw(ht); row_order(ht)'.

#K36
col_fun<- colorRamp2(c(-1, .2, quantile(K36_Heatmap_mat, 0.99)), c("white", "white", "brown"))
col_fun(seq(-1, quantile(K36_Heatmap_mat, 0.99), length = 20))

## [1] "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FFFFFF"
## [7] "#FFFFFF" "#FDF5F4FF" "#F8E5E2FF" "#F3D4CFFF" "#EDC3BDFF" "#E7B3ABFF"
## [13] "#E1A29AFF" "#D99289FF" "#D28178FF" "#CA7167FF" "#C16157FF" "#B85048FF"
## [19] "#AF3E39FF" "#A52A2AFF"

K36_hm <- Heatmap(K36_Heatmap_mat, col = col_fun, cluster_columns = F, show_row_names = FALSE, show_col

## Warning: The heatmap has not been initialized. You might have different results
## if you repeatedly execute this function, e.g. when row_km/column_km was
## set. It is more suggested to do as 'ht = draw(ht); row_order(ht)'.

#Kac versus input
col_fun<- colorRamp2(c(-1, .2, quantile(Kac_Heatmap_mat, 0.95)), c("white", "white", "purple"))
col_fun(seq(-1, quantile(Kac_Heatmap_mat, 0.99), length = 20))

## [1] "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FFFFFF"
## [7] "#FFFFFF" "#FFFFFF" "#F9EEFEFF" "#F0D5FDFF" "#E6BBFCFF" "#DBA2FAFF"
## [13] "#CF88F8FF" "#C26DF6FF" "#B450F3FF" "#A42CF1FF" "#A020FOFF" "#A020FOFF"
## [19] "#A020FOFF" "#A020FOFF"

```

```

Kac_hm <- Heatmap(Kac_Heatmap_mat, col = col_fun, cluster_columns = F, show_row_names = FALSE, show_col_nam

## Warning: The heatmap has not been initialized. You might have different results
## if you repeatedly execute this function, e.g. when row_km/column_km was
## set. It is more suggested to do as 'ht = draw(ht); row_order(ht)'.

#Kac versus WT
col_fun<- colorRamp2(c(-1, .2, quantile(KacVwt_Heatmap_mat, 0.95)), c("white", "white", "purple"))
col_fun(seq(-1, quantile(KacVwt_Heatmap_mat, 0.99), length = 20))

## [1] "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FEFBFFF" "#F8ECFEFF"
## [7] "#F3DCFEFF" "#EDCCFDFF" "#E6BCFCFF" "#EOADFBFF" "#D89DFAFF" "#D18DF9FF"
## [13] "#C97CF7FF" "#C16BF6FF" "#B85AF4FF" "#AF46F3FF" "#A52FF1FF" "#A020F0FF"
## [19] "#A020F0FF" "#A020F0FF"

Kac_Vwt_hm <- Heatmap(KacVwt_Heatmap_mat, col = col_fun, cluster_columns = F, show_row_names = FALSE, s

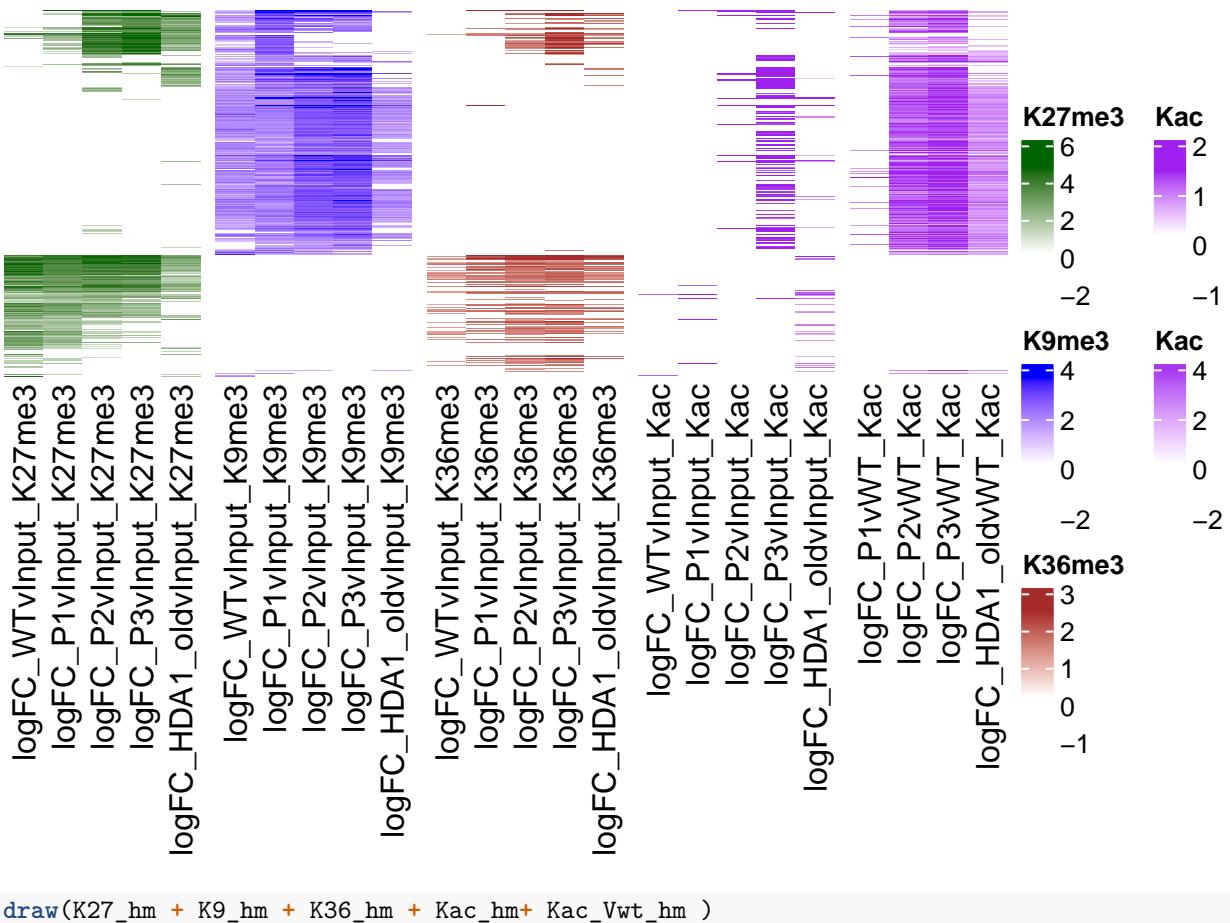
## Warning: The heatmap has not been initialized. You might have different results
## if you repeatedly execute this function, e.g. when row_km/column_km was
## set. It is more suggested to do as 'ht = draw(ht); row_order(ht)'.

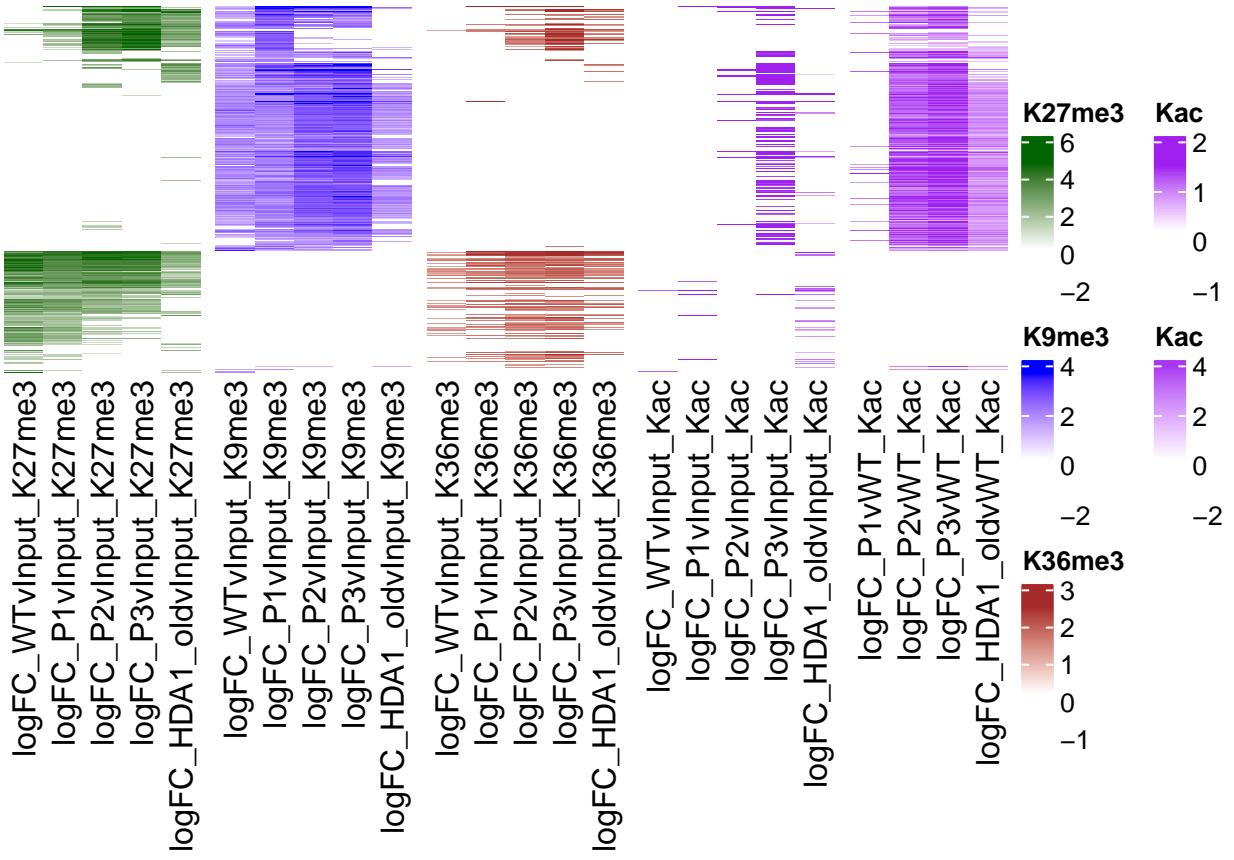
png("heatmap.png", width = 800, height = 800)
# Draw the heatmap
draw(K27_hm + K9_hm + K36_hm + Kac_hm+ Kac_Vwt_hm )
# Close the device to save the file
dev.off()

## pdf
## 2

K27_hm + K9_hm + K36_hm + Kac_hm+ Kac_Vwt_hm

```





```

# library(ggplot2)
#
# WT <- ggplot(NULL, aes(logFC_WTvInput_K9me3, logFC_WTvInput_K27me3)) +
#   geom_point(data = df1.K9peaksOnly, colour="blue") +
#   geom_point(data = df1.K27peaksOnly, colour="green")
#
# P1 <- ggplot(NULL, aes(logFC_P1vInput_K9me3, logFC_P1vInput_K27me3)) +
#   geom_point(data = df1.K9peaksOnly, colour="blue") +
#   geom_point(data = df1.K27peaksOnly, colour="green")
#
# P3<- ggplot(NULL, aes(logFC_P3vInput_K9me3, logFC_P3vInput_K27me3)) +
#   geom_point(data = df1.K9peaksOnly, colour="blue") +
#   geom_point(data = df1.K27peaksOnly, colour="green")
#
# WT
# P1
# P3
#
# WT <- ggplot(df1.K9peaksOnly, aes(logFC_WTvInput_K9me3, logFC_WTvInput_K27me3)) + geom_point(colour="blue")
#   geom_point(df1.K27peaksOnly, aes(logFC_WTvInput_K9me3, logFC_WTvInput_K27me3), color="green")
#   #geom_text_repel(data=l1, direction="both",fontface="italic",
#   #aes(log(NumberOfGenes_ZabovePointFive,2), MeanZ,label=paste("\u0394",locus, sep="")), min.segment.length=10)
#   # geom_point(data=caf, colour="red", size=3) +
#   # geom_point(data=cac3, colour="red", size=3) +
#   # geom_point(data=prc2, colour="blue", size=3) +
#   # geom_point(data=iswi, colour="#942850", size=3) +

```

```

#   # #geom_point(data=readers, colour="magenta")+
#   # geom_point(data=het, colour="#FDBB84", size=3)+
#   #geom_point(data=h2az, colour="#98D8CA")+
#   labs(x="Number of PRC2 target genes induced", y="Expression level of PRC2 target genes (n=623)")+
#   #geom_text_repel(data=l2, direction="both", fontface="italic",
#   #aes(log(NumberofGenes_ZabovePointFive,2), MeanZ, label=paste("\u0394", locus, sep="")), min.segment.length=1,
#   theme(panel.background = element_rect(color=NA, fill = "white"))+
#   theme(plot.background = element_rect(fill = "white"))+
#   theme(axis.line = element_line(color = "black",
#   linewidth = 1))+ 
#   theme(axis.ticks = element_blank())+
#   theme(axis.text = element_blank())+
#   theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
#
# ggsave("/Users/zacharylewis/Dropbox/DropBOX Documents/5. Zack presentations/2024/Fungal Genetics/caf_2024.pdf", width=10, height=8)
# 
```

# Notes:

#1 For scatter plot - pairwise comparisons should be made individually so that an FDR value generates

WT\_old.H3K27me3