Della Volpe et al

```
library(Seurat)
library(clusterProfiler)
library(dplyr)
library(viridis)
library(ggplot2)
library(grid)
library(gridExtra)
library(reshape2)
library(scales)
library(RColorBrewer)
library(ComplexHeatmap)
library(circlize)
library(stringr)
library(ggalluvial)
library(ggrepel)
library(openxlsx)
library(tidyr)
library(ggalluvial)
library(gtools)
library(cowplot)
#devtools::install_github("psyteachr/introdataviz")
library('introdataviz')
library(ggpubr)

full_obj <- readRDS(paste(wdir,"Full_final.rds", sep = "/")) #resolution 2
```

Figure 4 b - Umap annotated clusters

```
mycols <- c('HSC_MPP' = "lightblue1", 'MPP' = 'skyblue2',
'LMPP'="skyblue3",'LyP'="sandybrown",
            'GMP_1' = "lightgreen" ,'GMP_2'="#0CB702" ,'GMP_granulocyte'=
"olivedrab4",
            'MDP'= "#8494FF",'ProB'= "#E68613",'MEP'="lightpink",
            'MEMBP_1'= "indianred1", 'MEMBP_2'="indianred3" ,'EryP'="indianred4",
            'EBMP' = "#FF68A1","MEP_Baso_Mast"="#CD9600" )


Idents(full_obj) <- full_obj$Celltype
p <- DimPlot(full_obj, pt.size = 0.45, cols=mycols)
px <- LabelClusters(p, id = "ident", repel=T, color = "black",size=5)
+#+theme_void()
  theme(axis.text=element_text(size=10),
        axis.line = element_line(colour = "black", size=0.7),
        axis.title=element_text(size=22),
```

```
        axis.text.x = element_text(hjust=1,size=10),
        axis.text.y = element_text(size = 10),
        panel.grid = element_blank(),
        panel.border = element_blank(),
        text = element_text( size= 12)) +
  xlab('UMAP 1') + ylab('UMAP 2')


#svg(paste(out_dir,"Paper_plots", "umap_.svg", sep='/'), width =13, height = 10,
pointsize=12)
#px
#dev.off()
px
```

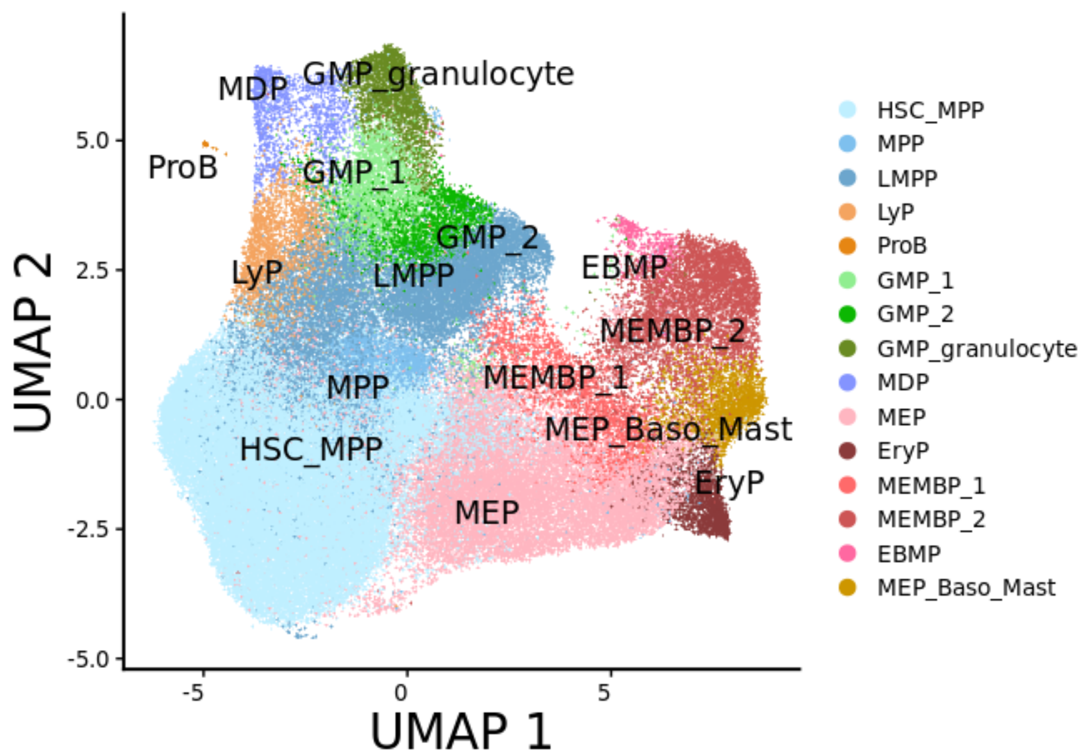


Figure 4 c - Barplot full object cell type cluster composition

```
table_cell_comp <- full_obj@meta.data %>%
  group_by(Celltype, RNA_pop) %>%
  count() %>%
  group_by(RNA_pop) %>%
  mutate(percent=100*n/sum(n)) %>%
  ungroup()
```

```
table_cell_comp$treatment <- ifelse(table_cell_comp$RNA_pop == 'UT_DMSO_pop','Not
edited',
                              ifelse(table_cell_comp$RNA_pop == 'UT_p38i_pop','Not
edited',
                              ifelse(table_cell_comp$RNA_pop == 'HS_DMSO_pop','HS
Cas9',
                              ifelse(table_cell_comp$RNA_pop == 'HS_p38i_pop', 'HS
Cas9',
                              ifelse(table_cell_comp$RNA_pop == 'GE_DMSO_pop_GFP-',
'GFP-',
ifelse(table_cell_comp$RNA_pop == 'GE_p38i_pop_GFP-', 'GFP-',
                              ifelse(table_cell_comp$RNA_pop == 'GE_DMSO_pop_GFP+',
'GFP+',
ifelse(table_cell_comp$RNA_pop == 'GE_p38i_pop_GFP+', 'GFP+',
                             'ND' ))))))))

table_cell_comp$treatment <- factor(table_cell_comp$treatment, levels=c(
  "Not edited", "HS Cas9",    "GFP-",       "GFP+"
))

table_cell_comp$Sample_name <- ifelse(table_cell_comp$RNA_pop ==
'UT_DMSO_pop','DMSO',
                                ifelse(table_cell_comp$RNA_pop ==
'UT_p38i_pop','p38i',
                                ifelse(table_cell_comp$RNA_pop ==
'HS_DMSO_pop','DMSO',
                                ifelse(table_cell_comp$RNA_pop == 'HS_p38i_pop',
'p38i',
                                ifelse(table_cell_comp$RNA_pop == 'GE_DMSO_pop_GFP-
', 'DMSO',
                                ifelse(table_cell_comp$RNA_pop == 'GE_p38i_pop_GFP-
', 'p38i',
ifelse(table_cell_comp$RNA_pop == 'GE_DMSO_pop_GFP+', 'DMSO',
                                ifelse(table_cell_comp$RNA_pop ==
'GE_p38i_pop_GFP+', 'p38i',
                               'ND' ))))))))



## save table
d_list <- list()
d_num <- dcast(table_cell_comp[,c(1,2,3)], RNA_pop ~ Celltype)
d_perc <- dcast(table_cell_comp[,c(1,2,4)], RNA_pop ~ Celltype)
d_list[["Number_Cells_CC_cluster"]] <- d_num
d_list[["Perc_Cells_CC_cluster"]] <- d_perc
## save perc and number
#out_dir <- 'Full/Paper_plots/tables'
#write.xlsx(x =d_list, file = paste(out_dir, 'Celltype_full_cell_perc.xlsx', sep =
"/"), rowNames = FALSE)

table_cell_comp$Celltype <- factor(table_cell_comp$Celltype, levels =
```

```r
rev(levels(table_cell_comp$Celltype)))

p1 <- table_cell_comp %>%
  ggplot(aes(x=Sample_name,y=percent, fill=Celltype)) +
  geom_col(aes(fill = Celltype), width = .5, color = "black")+
  #ggtitle("Percentage of cell cycle phases per cluster") +
  scale_fill_manual(name = '', values = mycols) +
  theme_classic() +
  scale_y_continuous(name = '% of cells (Alive + CD90+)', expand = c(0.01,0)) +
  geom_flow(aes(alluvium = Celltype),min.y= -1, alpha= .9,
            lty = 2, fill = "white", color = "black",
            curve_type = "linear",
            width = .5) +
  facet_wrap(~treatment,nrow=1, strip.position = "bottom") +
  theme(
    legend.position = 'right',
    plot.title = element_text(hjust = 0.5),
    text = element_text(size = 16),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.title.x = element_blank(),
    plot.margin = margin(t = 20, r = 10, b = 10, l = 10, unit = 'pt'),
    axis.line = element_line(colour = "black",
                             size = 0.3, linetype = "solid"),
    axis.text.x = element_text(size=12, color='black'), # angle=0, hjust=1,
    axis.text.y = element_text(size=12, color='black'),
    strip.placement = "outside")


#svg(paste(out_dir,"barplot_full_celltype_composition.svg", sep='/'), width =8,
height = 4, pointsize=12)
#p1
#dev.off()
p1
```

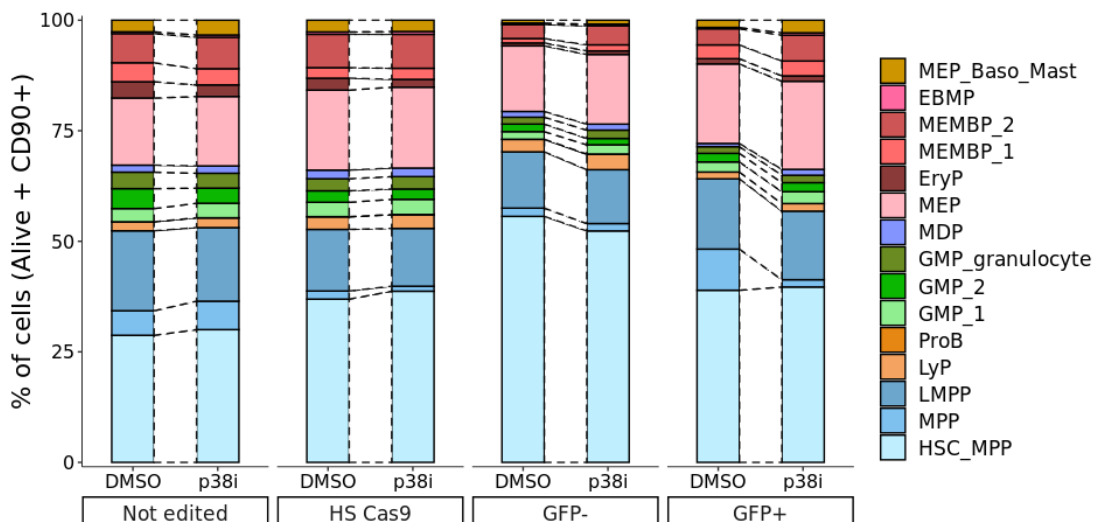


Figure 4 d - Barplot cell type cluster composition in sorted population CD34+CD133+CD90+CD45RA-

```
table_cell_comp <- full_obj@meta.data %>%
  group_by(Celltype, RNA_hashtag) %>%
  count() %>%
  group_by(RNA_hashtag) %>%
  mutate(percent=100*n/sum(n)) %>%
  ungroup()


table_cell_comp$type <-  unlist(lapply(str_split(table_cell_comp$RNA_hashtag,
'_'),"[", 3))
table_cell_comp <- table_cell_comp[table_cell_comp$type == 'CD90+',]
table_cell_comp$treatment <- ifelse(table_cell_comp$RNA_hashtag ==
"UT_p38i_CD90+",'Not edited',
                             ifelse(table_cell_comp$RNA_hashtag ==
"UT_DMSO_CD90+",'Not edited',
                             ifelse(table_cell_comp$RNA_hashtag ==
"HS_DMSO_CD90+",'HS Cas9',
                             ifelse(table_cell_comp$RNA_hashtag ==
"HS_p38i_CD90+", 'HS Cas9',
                             ifelse(table_cell_comp$RNA_hashtag ==
"GE_DMSO_CD90+_GFP-", 'GFP-',
                             ifelse(table_cell_comp$RNA_hashtag ==
"GE_p38i_CD90+_GFP-", 'GFP-',
                             ifelse(table_cell_comp$RNA_hashtag ==
"GE_DMSO_CD90+_GFP+",'GFP+',
                             ifelse(table_cell_comp$RNA_hashtag ==
"GE_p38i_CD90+_GFP+", 'GFP+',
                             'ND' ))))))))

table_cell_comp$treatment <- factor(table_cell_comp$treatment, levels=c(
  "Not edited", "HS Cas9",    "GFP-",       "GFP+"
))


table_cell_comp$Sample_name <- ifelse(table_cell_comp$RNA_hashtag ==
"UT_p38i_CD90+",'p38i',
                               ifelse(table_cell_comp$RNA_hashtag ==
"UT_DMSO_CD90+",'DMSO',
                               ifelse(table_cell_comp$RNA_hashtag ==
"HS_DMSO_CD90+",'DMSO',
                               ifelse(table_cell_comp$RNA_hashtag ==
"HS_p38i_CD90+", 'p38i',
                               ifelse(table_cell_comp$RNA_hashtag ==
"GE_DMSO_CD90+_GFP-", 'DMSO',
                               ifelse(table_cell_comp$RNA_hashtag ==
"GE_p38i_CD90+_GFP-", 'p38i',
```

```r
                              ifelse(table_cell_comp$RNA_hashtag ==
"GE_DMSO_CD90+_GFP+",'DMSO',
                              ifelse(table_cell_comp$RNA_hashtag ==
"GE_p38i_CD90+_GFP+", 'p38i',
                              'ND' ))))))))


## save table
d_list <- list()
d_num <- dcast(table_cell_comp[,c(1,2,3)], RNA_hashtag ~ Celltype)
d_perc <- dcast(table_cell_comp[,c(1,2,4)], RNA_hashtag ~ Celltype)
d_list[["Number_Cells_CC_cluster"]] <- d_num
d_list[["Perc_Cells_CC_cluster"]] <- d_perc
## save perc and number

#write.xlsx(x =d_list, file = paste(out_dir, 'Celltype_CD90_cell_perc.xlsx', sep =
"/"), rowNames = FALSE)

table_cell_comp_CD90 <- table_cell_comp
table_cell_comp_CD90$Celltype <- factor(table_cell_comp_CD90$Celltype, levels =
rev(levels(table_cell_comp_CD90$Celltype)))

p1 <- table_cell_comp_CD90 %>%
  ggplot(aes(x=Sample_name,y=percent, fill=Celltype)) +
  geom_col(aes(fill = Celltype), width = .5, color = "black")+
  #ggtitle("Percentage of cell cycle phases per cluster") +
  scale_fill_manual(name = '', values = mycols) +
  theme_classic() +
  scale_y_continuous(name = '% of CD90+ cells', expand = c(0.01,0)) +
  geom_flow(aes(alluvium = Celltype),min.y= -1, alpha= .9,
            lty = 2, fill = "white", color = "black",
            curve_type = "linear",
            width = .5) +
  facet_wrap(~treatment,nrow=1, strip.position = "bottom") +
  theme(
    legend.position = 'right',
    plot.title = element_text(hjust = 0.5),
    text = element_text(size = 16),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.title.x = element_blank(),
    plot.margin = margin(t = 20, r = 10, b = 10, l = 10, unit = 'pt'),
    axis.line = element_line(colour = "black",
                             size = 0.3, linetype = "solid"),
    axis.text.x = element_text(size=12, color='black'), # angle=0, hjust=1,
    axis.text.y = element_text(size=12, color='black'),
    strip.placement = "outside")


#svg(paste(out_dir,"barplot_cd90+_celltype_composition.svg", sep='/'), width =8,
```

```
height = 4, pointsize=12)
#p1
#dev.off()
p1
```

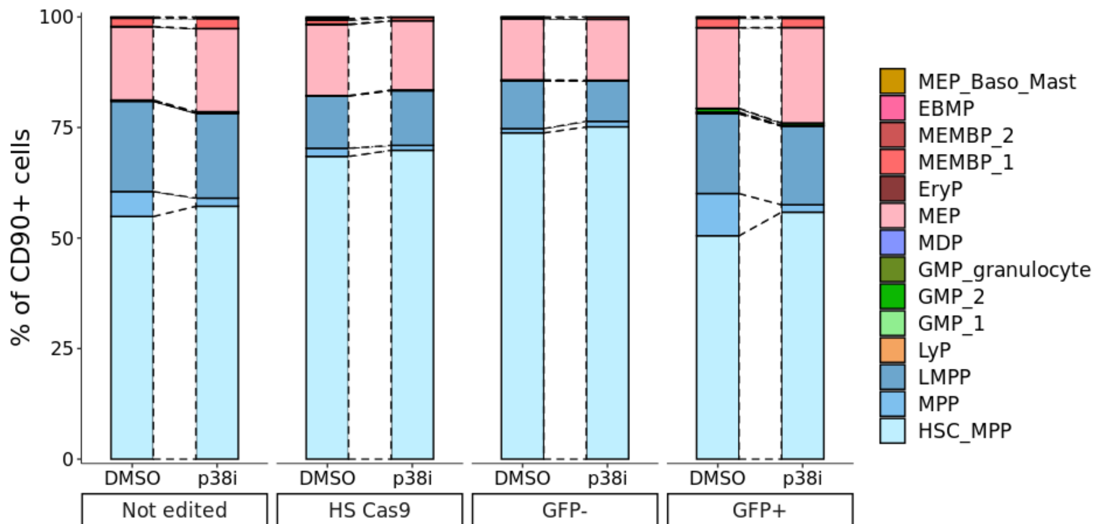


Figure 4 h - GSEA hsc signatures from literature Laurenti et al.

```
lset <- list('hsc' = c('RodriguezFraticelli_2020.gmt', 'Garcia-Prat2021.gmt'))

c <- list("UT_p38i_pop_vs_UT_DMSO_pop" = c("UT_p38i_pop","UT_DMSO_pop"),
          "HS_p38i_pop_vs_HS_DMSO_pop" = c( "HS_p38i_pop","HS_DMSO_pop"),
          "GE_DMSO_pop_GFP+_vs_GE_DMSO_pop_GFP-" =
c("GE_DMSO_pop_GFP+","GE_DMSO_pop_GFP-"), ## h3 vs h3
          "GE_p38i_pop_GFP+_vs_GE_DMSO_pop_GFP+" =
c("GE_p38i_pop_GFP+","GE_DMSO_pop_GFP+"), ## h4 vs h3
          "GE_p38i_pop_GFP-_vs_GE_DMSO_pop_GFP-" = c("GE_p38i_pop_GFP-
","GE_DMSO_pop_GFP-"),
          "GE_DMSO_pop_GFP+_vs_UT_DMSO_pop" = c("GE_DMSO_pop_GFP+","UT_DMSO_pop"),
## h3 vs h3
          "GE_DMSO_pop_GFP-_vs_UT_DMSO_pop" = c("GE_DMSO_pop_GFP-","UT_DMSO_pop"),
          "GE_p38i_pop_GFP+_vs_UT_DMSO_pop" = c("GE_p38i_pop_GFP+","UT_DMSO_pop"),
          "GE_p38i_pop_GFP-_vs_UT_DMSO_pop" = c("GE_p38i_pop_GFP-","UT_DMSO_pop"))
## h3 vs h3


x <- c('HSC_MPP')

comparisons <- expand.grid(x,names(c))
comp <- paste0(comparisons$Var1, '_', comparisons$Var2)
list_comp <- c()
gsea_res<-list()
```

```
out_dir <- 'Full/DGE_analysis/GSEA_alivecd90_together_HSC'
marker_dir <- 'Full/DGE_analysis/DGE_alivecd90_together/tables'
for (compx in comp) {  #for each comparisons within cluster
  for(nn in names(lset)){ # gmt list
    lset_name = lset[[nn]] ## get lists
    fi <- paste(out_dir, paste0(nn,'_',compx,"_GSEA.xlsx"), sep = "/")

    if (file.exists(fi)) { next }
    for (gset in lset_name) { #get specific gmt file

      t2gene <- read.gmt(gmtfile = paste0("reference/GSEA/",gset))

      ## check if file exist
      f <- paste(marker_dir, paste0(compx, '.txt'),sep = '/')
      if (file.exists(f)) {
        markers <- read.table(f)

        genelist <- markers$avg_log2FC
        names(genelist) <- rownames(markers) #$row.names
        # genelist <- na.omit(genelist)
        genelist <- sort(genelist, decreasing = T)

        print(paste0("Running GSEA ",compx, " - ", gset,'-',nn  ))


        ## do not run again
        #gsea_res[[nn]][[gset]][[compx]] <- GSEA(genelist, TERM2GENE = t2gene,
        #                                        maxGSSize = 1000, minGSSize =
10,seed = 123,
        #
        #                                        verbose = FALSE,
        #                                        pvalueCutoff = 1)

      }

      if (length(gsea_res[[nn]][[gset]][[compx]]) > 0){
        gsea_res[[nn]][[gset]][[compx]]@result['comparisons'] <- compx
        #write.xlsx(x =gsea_res[[nn]][[gset]][[compx]], file = paste(out_dir,
paste0(nn,'_',compx,"_GSEA.xlsx"), sep = "/"), rowNames = TRUE)
      }
    }
  }
}
#saveRDS(gsea_res, file = paste(out_dir, 'gsea_deg_gseaplot_HSC_mouse_human.rds',
sep = "/"))
gsea_res <- readRDS(paste(out_dir,
'gsea_deg_gseaplot_HSC_mouse_human.rds',sep='/'))

only_hsc <- c('HSC_MPP')
```

```r
c <- list("UT_p38i_pop_vs_UT_DMSO_pop" = c("UT_p38i_pop","UT_DMSO_pop"),
          "HS_p38i_pop_vs_HS_DMSO_pop" = c( "HS_p38i_pop","HS_DMSO_pop"),
          "GE_p38i_pop_GFP+_vs_GE_DMSO_pop_GFP+" =
c("GE_p38i_pop_GFP+","GE_DMSO_pop_GFP+"),
          "GE_p38i_pop_GFP-_vs_GE_DMSO_pop_GFP-" = c("GE_p38i_pop_GFP-
","GE_DMSO_pop_GFP-"))

comparisons <- expand.grid(only_hsc,names(c))

hallmark.full <- data.frame()
comp <- paste0(comparisons$Var1, '_', comparisons$Var2)
for (compx in comp) {
  for(nn in names(lset)){ # m,arco-group for instance hema, senescence in gmt list
    lset_name = lset[[nn]]
    for (gset in lset_name) {
       df <- gsea_res[[nn]][[gset]][[compx]]@result
       df$Category <- gset
       df$Dataset <- compx
      if (nrow(hallmark.full) == 0) {
         hallmark.full <- df
        } else {
        hallmark.full <- rbind(hallmark.full,df)
     }
    }
  }
}
#############
#write.table(hallmark.full.filt,paste(out_dir,'gsea_hsc_HUMAN_MOUSE.txt'))
hallmark.full.filt <- hallmark.full

hallmark.full.filt$p.adjust <- round(hallmark.full.filt$p.adjust)
hallmark.subset  <- subset(hallmark.full.filt , p.adjust <= 0.1)

min0 <- min(hallmark.subset[!is.na(hallmark.subset$NES),'NES'])
max0 <- max(hallmark.subset[!is.na(hallmark.subset$NES),'NES'])

hallmark.subset$stars=stars.pval(hallmark.subset$pvalue)

hallmark.order <- hallmark.subset%>% group_by(ID) %>% summarise(Pos = sum(NES))
hallmark.order.terms <- hallmark.order[order(hallmark.order$Pos, decreasing =
TRUE), "ID", drop = FALSE]
hallmark.subset$ID <- factor(hallmark.subset$ID, levels = hallmark.order.terms$ID)


hallmark.subset$Dataset <- factor(hallmark.subset$Dataset,
levels=c("HSC_MPP_UT_p38i_pop_vs_UT_DMSO_pop",

"HSC_MPP_HS_p38i_pop_vs_HS_DMSO_pop",
```

```
"HSC_MPP_GE_p38i_pop_GFP-_vs_GE_DMSO_pop_GFP-",

"HSC_MPP_GE_p38i_pop_GFP+_vs_GE_DMSO_pop_GFP+") )

hallmark.subset$Dataset <- ifelse(hallmark.subset$Dataset ==
"HSC_MPP_UT_p38i_pop_vs_UT_DMSO_pop", 'Not edited',
                            ifelse(hallmark.subset$Dataset
=="HSC_MPP_HS_p38i_pop_vs_HS_DMSO_pop", 'HS Cas9',
                            ifelse(hallmark.subset$Dataset
=="HSC_MPP_GE_p38i_pop_GFP-_vs_GE_DMSO_pop_GFP-", "GFP-",
ifelse(hallmark.subset$Dataset =="HSC_MPP_GE_p38i_pop_GFP+_vs_GE_DMSO_pop_GFP+",
"GFP+", 'none' ))))


hallmark.subset$Dataset <- factor(hallmark.subset$Dataset, levels=rev(c('Not
edited',
                                                                      'HS Cas9',
                                                                      "GFP-",
                                                                      "GFP+") ))
hallmark.subset$ID <- gsub('_', ' ', hallmark.subset$ID)
hallmark.subset$ID <- str_wrap(hallmark.subset$ID ,width = 30)
name_list <- list('Lauridsen et al 2019'=c("RACFP DIM Lauridsen et al"),
                  'Rodriguez et al 2020'=c("HSC1 cluster","HSC2 cluster","HSC3
cluster","HSC4 cluster",
                 "Low-output Rodriguez et al","High-output Rodriguez et al", "Mk-
bias Rodriguez et al","Serial-engraftment Rodriguez\net al", "Multilineage
Rodriguez et al"),
                  'Pietras et al 2014' = c("HSC Pietras et al"),
                  "Stemscore Giladi et al" = c("Stemscore Giladi et al"),
                  'Wilson et al' = c("SuMO Wilson et al", "MolO Wilson et al"),
                  'Cabezas-Wallscheid et al' = c("dHSC aHSC single-cell\nCabezas-
Wallscheid et al","HSC vs MPP4 Cabezas-Wallscheid\net al","dHSC vs aHSC\nCabezas-
Wallscheid et al"))

hallmark.subset$Name <- ifelse(hallmark.subset$ID %in% name_list$`Lauridsen et al
2019`, 'Lauridsen et al 2019',
                         ifelse(hallmark.subset$ID %in% name_list$`Rodriguez et al
2020`, 'Rodriguez et al 2020',
                         ifelse(hallmark.subset$ID %in% name_list$`Pietras et al
2014`, 'Pietras et al 2014',
                         ifelse(hallmark.subset$ID %in% name_list$`Stemscore Giladi
et al`,"Stemscore Giladi et al",
                         ifelse(hallmark.subset$ID %in% name_list$`Wilson et al`,
"Wilson et al",
                         ifelse(hallmark.subset$ID %in% name_list$`Cabezas-
Wallscheid et al`,'Cabezas-Wallscheid et al','Garcia-Prat'   ))))))

# exclude singatures
hallmark.subset <- hallmark.subset[!hallmark.subset$ID %in% c("HSC Pietras et al",
"Mk-bias Rodriguez et al","Serial-engraftment Rodriguez\net al",  "Multilineage
Rodriguez et al"),]
```

```
hallmark.subset$ID <- factor(hallmark.subset$ID,
levels=c(unname(unlist(name_list)), "Quiescent LT- vs ST-HSC",
                          "TFEB vs CTRL MEP","TFEB vs CTRL ST-HSC", "LT-HSC TFEBvsCTRL
RNAseq","MYC vs CTRL LT-HSC"))

plot1 <- ggplot(data = hallmark.subset, mapping = aes(x = Dataset, y = ID))
+#reorder(ID, NES))) +
  geom_tile(aes(fill=NES, group = Category), width=0.70, height=0.70)+
#80#color="white", size = 0.30) + labs(fill = "NES") +
  facet_grid(. ~ Category + Name, scales = "free", space = "free")+
  theme(text= element_text(size=22,color='black'),
        legend.position = 'top',
        axis.text.x = element_text(angle = 45, vjust = 1, hjust =1, color='black',
size =8),
        axis.text.y = element_text(color='black', size =10),
        plot.title = element_text(hjust = 0), plot.subtitle = element_text(hjust =
0),
        axis.ticks = element_blank(), axis.text = element_text(size = 8),
axis.title = element_blank(),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_rect(fill = "white", color='black'),
        strip.background = element_rect(colour="white",
                                        fill="white",size=0.5),
        strip.text = element_text(color='black',face = 'bold', size = 11)) +
  scale_fill_gradientn(colours =
colorRampPalette(rev(brewer.pal(11,"RdBu")))(100),
                       limits = c(min0,max0),
                       na.value = "grey")+ ggtitle('p38i vs DMSO') +coord_flip()


#wdir='Full/DGE_analysis/GSEA_alivecd90_together_HSC'
#svg(paste(wdir,"HSC_human_mouse_2.svg",sep = '/'),width=8,
#    height=5,   pointsize=12)
#plot1
#dev.off()
plot1
```

p38i vs DMSO

Figure 4 e - GSEA hallmark & GO BP

```
hallmark.subset <- read.table('Full/Paper_plots/tables/Main_Figure_4e.txt' )

min0 <- min(hallmark.subset[!is.na(hallmark.subset$NES),'NES'])
max0 <- max(hallmark.subset[!is.na(hallmark.subset$NES),'NES'])


hallmark.subset$stars=stars.pval(hallmark.subset$pvalue)

## adjust name too long
val <- "regulation of transcription involved in G1/S transition of mitotic cell
cycle (GO:0000083)"
valx <- "reg. transcr. involved in G1/S transition (GO:0000083)"
hallmark.subset$ID <- ifelse(hallmark.subset$ID == val, valx, hallmark.subset$ID)
hallmark.subset$Name <- ifelse(hallmark.subset$Category == 'go',
sapply(strsplit(as.character(hallmark.subset$ID), " \\("), `[`, 1),
as.character(hallmark.subset$ID))
hallmark.subset$ID <- str_wrap(hallmark.subset$ID ,width = 45)
hallmark.subset$celltype <- gsub('_', ' ', hallmark.subset$celltype)
hallmark.subset$celltype <- ifelse(hallmark.subset$celltype == 'MEP Baso Mast',
'MEP Baso-Mast', hallmark.subset$celltype)
#hallmark.subset$celltype <- ifelse(nchar(hallmark.subset$celltype) > 7,
str_wrap(hallmark.subset$celltype ,width = 6), hallmark.subset$celltype)

hallmark.subset$celltype <- factor(hallmark.subset$celltype, levels =
                                    c("HSC MPP",'MPP',"LMPP","GMP 1",
                                      "GMP 2","GMP\ngranulocyte",
"MDP","MEP","EryP","MEMBP 1",                                       "MEMBP 2",
"MEP\nBaso-Mast","LyP","ProB","EBMP" ))

hallmark.order <- hallmark.subset%>% group_by(ID) %>% summarise(Pos = sum(NES))
hallmark.order.terms <- hallmark.order[order(hallmark.order$Pos, decreasing =
TRUE), "ID", drop = FALSE]
hallmark.subset$ID <- factor(hallmark.subset$ID, levels = hallmark.order.terms$ID)


hallmark.subset$Dataset <- factor(hallmark.subset$Dataset,
levels=c("UT_p38i_pop_vs_UT_DMSO_pop",

"HS_p38i_pop_vs_HS_DMSO_pop",

"GE_p38i_pop_GFPneg_vs_GE_DMSO_pop_GFPneg",

"GE_p38i_pop_GFPpos_vs_GE_DMSO_pop_GFPpos"))

hallmark.subset$Dataset <- ifelse(hallmark.subset$Dataset ==
"UT_p38i_pop_vs_UT_DMSO_pop", 'Not edited',
                              ifelse(hallmark.subset$Dataset
```

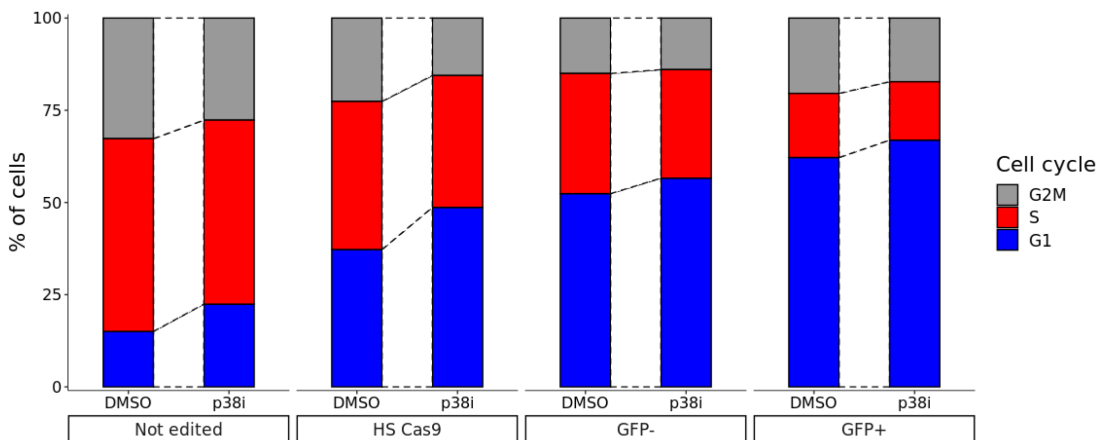
```
=="HS_p38i_pop_vs_HS_DMSO_pop", 'HS Cas9',
                           ifelse(hallmark.subset$Dataset
=="GE_p38i_pop_GFPneg_vs_GE_DMSO_pop_GFPneg", "GFP-",
                           ifelse(hallmark.subset$Dataset
=="GE_p38i_pop_GFPpos_vs_GE_DMSO_pop_GFPpos", "GFP+", 'none' ))))

hallmark.subset$Dataset <- factor(hallmark.subset$Dataset, levels=c('Not edited',
                                                                   'HS Cas9',
                                                                   "GFP-",
                                                                   "GFP+") )

hallmark.subset <- hallmark.subset[hallmark.subset$Category != 'p38_signature',]
p <- ggplot(data = hallmark.subset, mapping = aes(x = Dataset, y = reorder(Name, -
NES))) +
  geom_tile(aes(fill=NES, group = celltype), width=0.80, height=0.80)+
  facet_grid(Category ~ celltype  , scales = "free_y", space = "free", )+
  theme(text= element_text(size=22,color='black'),
        axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1,
color='black', size =12),
        axis.text.y = element_text(color='black', size =16),
        plot.title = element_text(hjust = 0), plot.subtitle = element_text(hjust =
0),
        axis.ticks = element_blank(), axis.text = element_text(size = 8),
axis.title = element_blank(),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_rect(fill = "gray95", color='black'),
        strip.background = element_rect(colour="white",
                                        fill="white",size=0.5),
        strip.text = element_text(color='black',face = 'bold', size = 11),
        legend.position = 'top', legend.direction = 'horizontal') +
    scale_fill_gradientn(colours =
colorRampPalette(rev(brewer.pal(11,"RdBu")))(100),
                         limits = c(min0,max0),
                         na.value = "grey")+ ggtitle('p38i vs DMSO')

#wdir='Full/DGE_analysis'
#svg(paste(wdir,"HM_categories_v4.svg",sep = '/'),width=22,
#   height=11,   pointsize=12)
#p
#dev.off()
p
```

Figure 4 f - cell cycle full object & per cluster

```
full_obj$RNA_pop <- factor( full_obj$RNA_pop,
levels=c("UT_DMSO_pop","UT_p38i_pop",
                                                       "HS_DMSO_pop"
,"HS_p38i_pop",

"GE_DMSO_pop_GFP+","GE_p38i_pop_GFP+",
                                                       "GE_DMSO_pop_GFP-",
"GE_p38i_pop_GFP-"))
table_cell_cycle <- full_obj@meta.data %>%
  group_by(RNA_pop,Phase) %>%
  count() %>%
  group_by(RNA_pop) %>%
  mutate(percent=100*n/sum(n)) %>%
  ungroup()

d_list <- list()
d_num <- dcast(table_cell_cycle[,c(1,2,3)], Phase ~ RNA_pop)
d_perc <- dcast(table_cell_cycle[,c(1,2,4)], Phase ~ RNA_pop)
d_list[["Number_Cells_CC_treatment"]] <- d_num
```

```
d_list[["Perc_Cells_CC_treatment"]] <- d_perc

#write.xlsx(x =d_list, file = paste(out_dir, 'Cellcycle_treatment.xlsx', sep =
"/"), rowNames = FALSE)

custom_colors <- list()
custom_colors$cell_cycle <- setNames(
  c("blue", "red", "gray60"), #'#7f8c8d',
  c('G1',      'S',       'G2M') #     '-')
)

table_cell_cycle$Phase <- factor(table_cell_cycle$Phase, levels =
c("G2M","S","G1"))
table_cell_cycle$treatment <- ifelse(table_cell_cycle$RNA_pop ==
'UT_DMSO_pop','Not edited',
                              ifelse(table_cell_cycle$RNA_pop ==
'UT_p38i_pop','Not edited',
                              ifelse(table_cell_cycle$RNA_pop == 'HS_DMSO_pop','HS
Cas9',
                              ifelse(table_cell_cycle$RNA_pop == 'HS_p38i_pop',
'HS Cas9',
                              ifelse(table_cell_cycle$RNA_pop == 'GE_DMSO_pop_GFP-
', 'GFP-',
                              ifelse(table_cell_cycle$RNA_pop == 'GE_p38i_pop_GFP-
', 'GFP-',
                              ifelse(table_cell_cycle$RNA_pop ==
'GE_DMSO_pop_GFP+', 'GFP+',
                              ifelse(table_cell_cycle$RNA_pop ==
'GE_p38i_pop_GFP+', 'GFP+', 'ND' ))))))))

table_cell_cycle$treatment <- factor(table_cell_cycle$treatment, levels=c(
  "Not edited", "HS Cas9",    "GFP-",       "GFP+"))

table_cell_cycle$Sample_name <- ifelse(table_cell_cycle$RNA_pop ==
'UT_DMSO_pop','DMSO',
                                ifelse(table_cell_cycle$RNA_pop ==
'UT_p38i_pop','p38i',
                                ifelse(table_cell_cycle$RNA_pop ==
'HS_DMSO_pop','DMSO',
                                ifelse(table_cell_cycle$RNA_pop == 'HS_p38i_pop',
'p38i',
                                ifelse(table_cell_cycle$RNA_pop ==
'GE_DMSO_pop_GFP-', 'DMSO',
                                ifelse(table_cell_cycle$RNA_pop ==
'GE_p38i_pop_GFP-', 'p38i',
                                ifelse(table_cell_cycle$RNA_pop ==
'GE_DMSO_pop_GFP+', 'DMSO',
                                ifelse(table_cell_cycle$RNA_pop ==
'GE_p38i_pop_GFP+', 'p38i',
                                'ND' ))))))))
```

```r
p1 <- table_cell_cycle %>%
  ggplot(aes(x=Sample_name,y=percent, fill=Phase)) +
  geom_col(aes(fill = Phase), width = .5, color = "black")+
  #ggtitle("Percentage of cell cycle phases per cluster") +
  scale_fill_manual(name = 'Cell cycle', values = custom_colors$cell_cycle) +
  theme_classic() +
  scale_y_continuous(name = '% of cells', expand = c(0.01,0)) +
  geom_flow(aes(alluvium = Phase),min.y= -1, alpha= .9,
            lty = 2, fill = "white", color = "black",
            curve_type = "linear",
            width = .5) +
  facet_wrap(~treatment,nrow=1, strip.position = "bottom") +
  theme(
    legend.position = 'right',
    plot.title = element_text(hjust = 0.5),
    text = element_text(size = 16),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.title.x = element_blank(),
    plot.margin = margin(t = 20, r = 10, b = 10, l = 10, unit = 'pt'),
    axis.line = element_line(colour = "black",
                             size = 0.3, linetype = "solid"),
    axis.text.x = element_text(size=12, color='black'), # angle=0, hjust=1,
    axis.text.y = element_text(size=12, color='black'),
    strip.placement = "outside",
    )

#svg(paste(out_dir,"CC_treatment_perc_colorcode.svg", sep='/'), width =8, height =
4, pointsize=12)
#
#p1
# dev.off()
p1
```

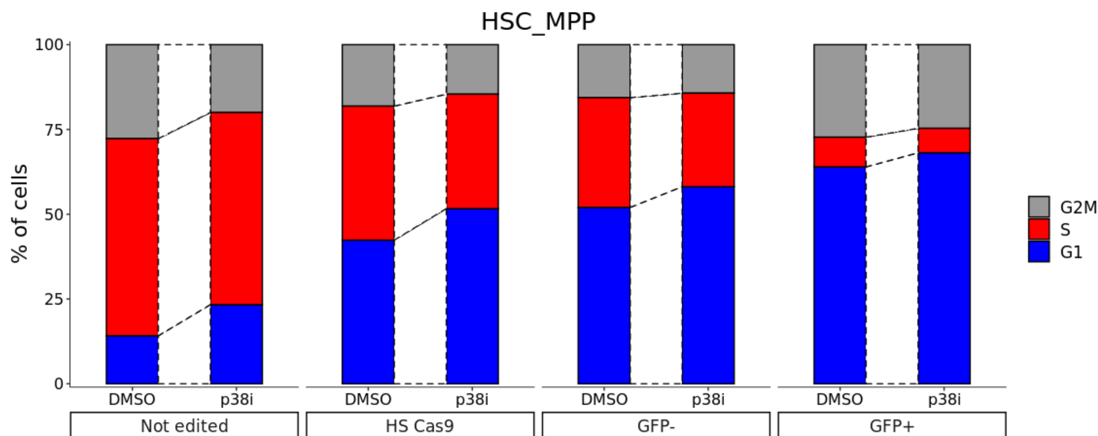



```r
##################################################
# cell cycle HSC cluster and all the others clusters
##################################################
```

```
table_cell_cycle <- full_obj@meta.data %>%
  group_by(Celltype, Phase, RNA_pop) %>%
  count() %>%
  group_by(RNA_pop,Celltype) %>%
  mutate(percent=100*n/sum(n)) %>%
  ungroup()

d_list <- list()
d_num <- dcast(table_cell_cycle[,c(1,2,3,4)], Celltype + Phase ~ RNA_pop )
d_perc <- dcast(table_cell_cycle[,c(1,2,3,5)], Celltype + Phase ~ RNA_pop)
d_list[["Number_Cells_CC_treatment"]] <- d_num
d_list[["Perc_Cells_CC_treatment"]] <- d_perc
out_dir <- 'Full/CellCycle'
#write.xlsx(x =d_list, file = paste(out_dir, 'Cellcycle_treatment_xcelltype.xlsx',
sep = "/"), rowNames = FALSE)

table_cell_cycle$Phase <- factor(table_cell_cycle$Phase, levels =
c("G2M","S","G1"))
table_cell_cycle$treatment <- ifelse(table_cell_cycle$RNA_pop ==
'UT_DMSO_pop','Not edited',
                                     ifelse(table_cell_cycle$RNA_pop ==
'UT_p38i_pop','Not edited',
                                            ifelse(table_cell_cycle$RNA_pop ==
'HS_DMSO_pop','HS Cas9',
                                                   ifelse(table_cell_cycle$RNA_pop
== 'HS_p38i_pop', 'HS Cas9',

ifelse(table_cell_cycle$RNA_pop == 'GE_DMSO_pop_GFP-', 'GFP-',

ifelse(table_cell_cycle$RNA_pop == 'GE_p38i_pop_GFP-', 'GFP-',

ifelse(table_cell_cycle$RNA_pop == 'GE_DMSO_pop_GFP+', 'GFP+',

ifelse(table_cell_cycle$RNA_pop == 'GE_p38i_pop_GFP+', 'GFP+',

'ND' ))))))))

table_cell_cycle$treatment <- factor(table_cell_cycle$treatment, levels=c(
  "Not edited", "HS Cas9",    "GFP-",       "GFP+"
))

table_cell_cycle$Sample_name <- ifelse(table_cell_cycle$RNA_pop ==
'UT_DMSO_pop','DMSO',
                                ifelse(table_cell_cycle$RNA_pop ==
'UT_p38i_pop','p38i',
                                ifelse(table_cell_cycle$RNA_pop ==
'HS_DMSO_pop','DMSO',
                                ifelse(table_cell_cycle$RNA_pop == 'HS_p38i_pop',
'p38i',
                                ifelse(table_cell_cycle$RNA_pop ==
'GE_DMSO_pop_GFP-', 'DMSO',
```

```
                                 ifelse(table_cell_cycle$RNA_pop ==
'GE_p38i_pop_GFP-', 'p38i',
                                 ifelse(table_cell_cycle$RNA_pop ==
'GE_DMSO_pop_GFP+', 'DMSO',
                                 ifelse(table_cell_cycle$RNA_pop ==
'GE_p38i_pop_GFP+', 'p38i',
                                                                                       'ND'
))))))))
p <- list()
for (n in unique(table_cell_cycle$Celltype)) {
  df <- data.frame(table_cell_cycle)
  dfx <- df[df$Celltype == as.character(n), ]
  dfx$roundfreq <- round(dfx$percent,2)

p[[n]] <- ggplot( dfx, aes(Sample_name, percent))+
  geom_col(aes(fill = Phase), width = .5, color = "black")+
  theme_classic() +
  #ggtitle("Percentage of cell cycle phases per cluster") +
  scale_fill_manual(name = '', values = custom_colors$cell_cycle) +
  theme_classic() +
  geom_flow(aes(alluvium = Phase),min.y= -1, alpha= .9,
            lty = 2, fill = "white", color = "black",
            curve_type = "linear",
            width = .5) + ggtitle(n)+
  scale_y_continuous(expand = c(0, 0))+
  facet_wrap(~treatment,nrow=1, strip.position = "bottom") +
  theme(
    legend.position = 'right',
    plot.title = element_text(hjust = 0.5),
    text = element_text(size = 16),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.title.x = element_blank(),
    plot.margin = margin(t = 20, r = 10, b = 10, l = 10, unit = 'pt'),
    axis.line = element_line(colour = "black",
                             size = 0.3, linetype = "solid"),
    axis.text.x = element_text(size=12, color='black'), # angle=0, hjust=1,
    axis.text.y = element_text(size=12, color='black'),
    #axis.title.y = element_blank(),
    strip.placement = "outside") +  scale_y_continuous(name = '% of cells', expand
= c(0.01,0))

}

#barplot_pz <- grid.arrange(grobs =
list(p$HSC_MPP,p$MPP,p$LMPP,p$LyP,p$ProB,p$GMP_1,p$GMP_2,p$GMP_granulocyte,p$MDP,p
$MEP,p$EryP,p$MEMBP_1,p$MEMBP_2,p$EBMP,p$MEP_Baso_Mast),
#              left = textGrob("", rot = 90, vjust = 1),ncol=4)

#leg <- get_legend(barplot_pz) ## extract the legend begore excluding it from plot
#p$leg <- leg
```

```
#svg(paste(out_dir,"CC_HSC_mpp_barplot_cellcycle_colorcoded.svg",sep =
'/'),width=8,
#    height=4,    pointsize=12)

#dev.off()
grid.arrange(grobs = list(p$HSC_MPP),
             ncol=1)
```



Supplementary Figures

Supplementary Figure 4 d

```
#svg(paste(out_dir,"barplot_cellcycle_colorcoded.svg",sep = '/'),width=10,
#    height=60,    pointsize=12)
#grid.arrange(grobs =
list(p$MPP,p$LMPP,p$LyP,p$ProB,p$GMP_1,p$GMP_2,p$GMP_granulocyte,p$MDP,p$MEP,p$Ery
P,p$MEMBP_1,p$MEMBP_2,p$EBMP,p$MEP_Baso_Mast),
#              ncol=4)
#dev.off()

grid.arrange(grobs =
list(p$MPP,p$LMPP,p$LyP,p$ProB,p$GMP_1,p$GMP_2,p$GMP_granulocyte,p$MDP,p$MEP,p$Ery
P,p$MEMBP_1,p$MEMBP_2,p$EBMP,p$MEP_Baso_Mast), ncol=4)
```

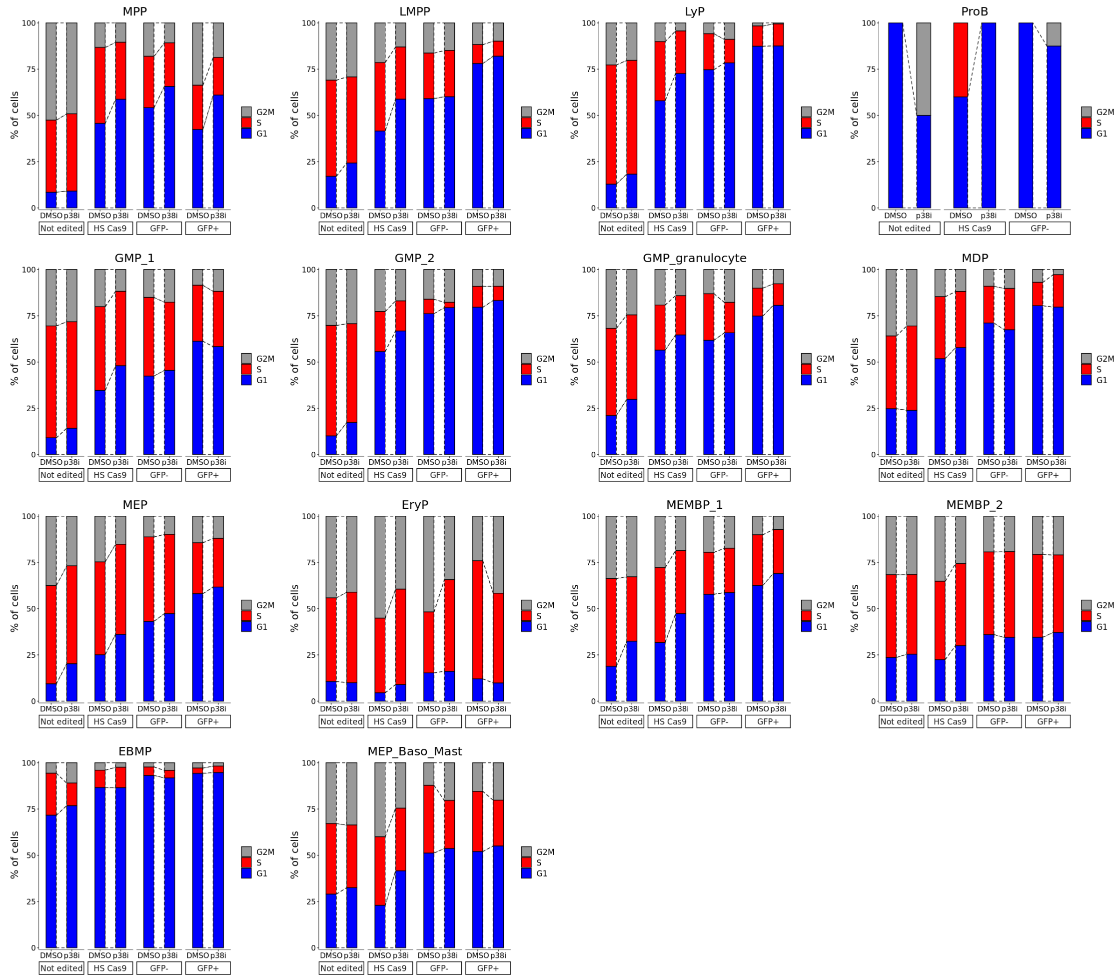


Figure 4 g - Lopes lists HSC-MPP

```
gmt.obj<- list(
  'Response_to_DNA_damage_replication_stress' =
c('Plk1','Trex1','Atrip','Trp53','Tipin','Timeless','Rpa2','Rfwd3','Cdk1','Taok3',
'Fzr1','Brcc3','Rad17','Babam2','Rad9a','Rad1','Cdc5l','Etaa1',
'Dtl','Chek1','Clspn','Hus1','Ticrr','Prpf19','Babam1','Tiprl','Abraxas1','Thoc1',
'Topbp1','Donson','Chek2','Map3k20','Nek11','Mdm2','Pidd1',
'Atr','Rhno1','Uimc1','Rbm38','Usp28','Atm','Trim39','Rad9b','Hus1b','Brsk1'),
'DNA_replication_inition'=
c('Ccne1','Gmnc','Cdc45','Mcm10','Cdc6','Mcm2','Mcm6','Mcm7','Mcm4','Mcm5','Mcm3',
'Cdt1','Noc3l','Orc5','Orc3','Kat7','Orc1','Mcidas','Wrnip1','Ccne2','Pola1','Gins
3','Pola2','Orc6','Prim2','Ticrr','Orc4','Prim1','Topbp1','Cdk2','Ciz1'))
ll <- gmt.obj #unique(gmt.obj$term)

## do not run again
#for (l in names(ll)) {
#  group <- toupper(ll[[l]]) #levels(gmt.obj$term)[l]
#  nn <- gsub('\\)','',gsub('\\(','',gsub(' ','_',l)))
```

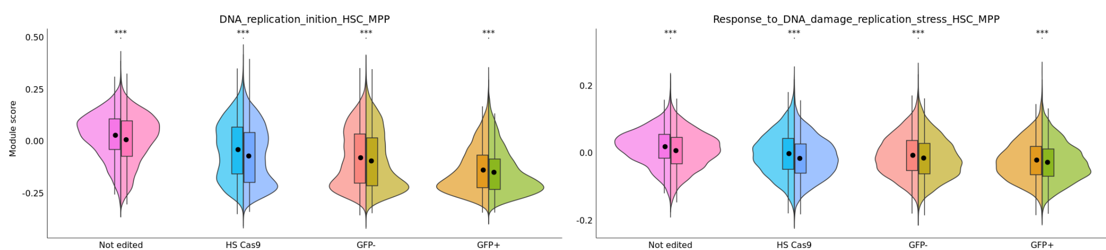
```r
#  gmt_features <- list(group)
#  full_obj <- AddModuleScore(
#    object = full_obj,
#    features = gmt_features, ##
#    name = nn,
#  )}


namex <- list("DNA_replication_inition" = "DNA replication inition",
              "Response_to_DNA_damage_replication_stress" = "Response to DNA
damage replication stress")


p1 <- list()
p <- list()
melt_005df <- list()
melt_005pp <- list()
meltpp <- list()
pp <- list()
legend = ''

Idents(full_obj) <- 'Celltype'


#subset <- subset(full_obj, idents='HSC_MPP')
out.dir <- 'Full'
#saveRDS(subset, paste(out.dir, 'HSC_MPP_cluster.rds', sep ='/'))

subset <- readRDS(paste(out.dir, 'HSC_MPP_cluster.rds', sep ='/'))

check_fun <- function(x,df){
  df$comparison <- 'comp'
  for (v in 1:dim(df)[[1]]) {
     d1 <- as.character(x[x$group == as.character(df$Var1[v]), 'treatment']) #
RNA_pop == group
     d2 <- as.character(x[x$group == as.character(df$Var2[v]), 'treatment'])#
RNA_pop == group
     if (d1 == d2) {
         ## keep value in the matrix
       df$comparison <- replace(df$comparison,v,d1)
     }

  }
  return(df)
}

for (i in names(namex)) {
  n <- paste0(i,'1')
```

```r
  antibio <- data.frame(subset@meta.data[[n]], subset@meta.data$RNA_pop)
  antibio$celltype <- 'HSC_MPP'

  i <- paste0(i, '_', 'HSC_MPP')
  colnames(antibio) <- c('value', 'RNA_pop', 'celltype')
  antibio$treatment <- ifelse(antibio$RNA_pop == 'UT_DMSO_pop','Not edited',
                                ifelse(antibio$RNA_pop == 'UT_p38i_pop','Not
edited',
                                ifelse(antibio$RNA_pop == 'HS_DMSO_pop','HS Cas9',
                                ifelse(antibio$RNA_pop == 'HS_p38i_pop', 'HS
Cas9',
                                ifelse(antibio$RNA_pop == 'GE_DMSO_pop_GFP-',
'GFP-',
                                ifelse(antibio$RNA_pop == 'GE_p38i_pop_GFP-',
'GFP-',
                                ifelse(antibio$RNA_pop == 'GE_DMSO_pop_GFP+',
'GFP+',
                                ifelse(antibio$RNA_pop == 'GE_p38i_pop_GFP+',
'GFP+',
                                'ND' ))))))))
  antibio$treatment <- factor(antibio$treatment, levels=c(
    "Not edited", "HS Cas9",    "GFP-",       "GFP+"
  ))

  antibio$Sample_name <- ifelse(antibio$RNA_pop == 'UT_DMSO_pop','DMSO',
                                  ifelse(antibio$RNA_pop == 'UT_p38i_pop','p38i',
                                  ifelse(antibio$RNA_pop == 'HS_DMSO_pop','DMSO',
                                  ifelse(antibio$RNA_pop == 'HS_p38i_pop', 'p38i',
                                  ifelse(antibio$RNA_pop == 'GE_DMSO_pop_GFP-',
'DMSO',
                                  ifelse(antibio$RNA_pop == 'GE_p38i_pop_GFP-',
'p38i',
                                  ifelse(antibio$RNA_pop == 'GE_DMSO_pop_GFP+',
'DMSO',
                                  ifelse(antibio$RNA_pop == 'GE_p38i_pop_GFP+',
'p38i',
                                  'ND' ))))))))

  dim(antibio)
  alpha_genes <-  0.05
  rownames(antibio) <- rownames(subset@meta.data)
  colnames(antibio) <- c('value','group','celltype', 'treatment', 'Sample_name')
## module score and RNA_pop
  antibio$group <-  factor( antibio$group ,
levels=c("UT_DMSO_pop","UT_p38i_pop","HS_DMSO_pop"
,"HS_p38i_pop","GE_DMSO_pop_GFP+","GE_p38i_pop_GFP+", "GE_DMSO_pop_GFP-",
"GE_p38i_pop_GFP-"))

  pp[[i]]<-list(pairwise.wilcox.test(antibio$value,antibio$group, p.adjust.method
= "BH", paired = FALSE))
  meltpp[[i]] <- data.frame(melt(pp[[i]][[1]]$p.value))
```

```
  meltpp[[i]][is.na(meltpp[[i]]$value),3] <- 100 # substitute na third column
  if (min(meltpp[[i]]$value) <= alpha_genes){
    melt_005pp[[i]] <-  filter(meltpp[[i]] , value <= alpha_genes)

    melt_005df[[i]] <- melt_005pp[[i]][complete.cases(melt_005pp[[i]]),]
    y_position <- list()
    for (y in 1:dim(melt_005df[[i]])[1]) {
      x <- as.numeric(paste(c('0.',y), collapse = ""))
      y_position[y] <- as.numeric(max(antibio$value)+0.1)
    }
    melt_005df[[i]]$y.position <- y_position
    melt_005df[[i]]$p.signif<- symnum(melt_005df[[i]]$value, cutpoints = c(0,
0.0001, 0.001, alpha_genes, 1),
                                      symbols = c( "***", "**", "*", "ns"))

    df <- melt_005df[[i]]
    annotation <- unique(antibio[,c('treatment', 'group')])
    df_x <-  check_fun(annotation,df)
    melt_005df[[i]] <- filter(df_x , comparison != "comp")
    if( nrow(melt_005df[[i]]) > 0) {
    ## substitute name

    melt_005df[[i]]$Var1_mod  <- ifelse(melt_005df[[i]]$Var1 ==
'UT_DMSO_pop','DMSO',
                                 ifelse(melt_005df[[i]]$Var1 ==
'UT_p38i_pop','p38i',
                                 ifelse(melt_005df[[i]]$Var1 ==
'HS_DMSO_pop','DMSO',
                                 ifelse(melt_005df[[i]]$Var1 == 'HS_p38i_pop',
'p38i',
                                 ifelse(melt_005df[[i]]$Var1 == 'GE_DMSO_pop_GFP-
', 'DMSO',
                                 ifelse(melt_005df[[i]]$Var1 == 'GE_p38i_pop_GFP-
', 'p38i',
                                 ifelse(melt_005df[[i]]$Var1 ==
'GE_DMSO_pop_GFP+', 'DMSO',
                                 ifelse(melt_005df[[i]]$Var1 ==
'GE_p38i_pop_GFP+', 'p38i',
                                 'ND' ))))))))

    melt_005df[[i]]$Var2_mod  <- ifelse(melt_005df[[i]]$Var2 ==
'UT_DMSO_pop','DMSO',
                                 ifelse(melt_005df[[i]]$Var2 ==
'UT_p38i_pop','p38i',
                                 ifelse(melt_005df[[i]]$Var2 ==
'HS_DMSO_pop','DMSO',
                                 ifelse(melt_005df[[i]]$Var2 == 'HS_p38i_pop',
'p38i',
                                 ifelse(melt_005df[[i]]$Var2 == 'GE_DMSO_pop_GFP-
', 'DMSO',
                                 ifelse(melt_005df[[i]]$Var2 == 'GE_p38i_pop_GFP-
```

```
', 'p38i',
                                  ifelse(melt_005df[[i]]$Var2 ==
'GE_DMSO_pop_GFP+', 'DMSO',
                                  ifelse(melt_005df[[i]]$Var2 ==
'GE_p38i_pop_GFP+', 'p38i',
                                  'ND' ))))))))

    ## comparison df, create table
    comparisons <- data.frame(
      .y. = c(rep('len',dim(melt_005df[[i]])[1])),
      group1 = melt_005df[[i]]$Var1_mod, #_mod,
      group2 = melt_005df[[i]]$Var2_mod, #_mod,
      p = melt_005df[[i]]$value,
      p.adj = round(melt_005df[[i]]$value,3),
      p.format = melt_005df[[i]]$value,
      p.signif = melt_005df[[i]]$p.signif,
      method = 'Wc',
      y.position = unlist(melt_005df[[i]]$y.position),
      group = unlist(melt_005df[[i]]$comparison),
      g1 = melt_005df[[i]]$Var1,
      g2 =melt_005df[[i]]$Var2
    ) %>%
      as.tbl()
    #write.table(out_dir, 'DGE_analysis', 'Plot_Lopes', paste0(i, '.txt'),
sep='/'))

    antibio$tr <- antibio$group
    p1[[i]]  <- ggplot(antibio, aes(x = treatment, y = value, fill=group))+
      geom_split_violin(alpha = .6, trim = FALSE, aes(fill=group)) +
      geom_boxplot(aes(fill=group),   outlier.size=0,outlier.alpha =0,
                   width = .2, alpha = .7, fatten = NULL, show.legend = FALSE) +
      geom_signif(
         data=comparisons,
         aes(xmin=group, xmax=group, annotations=p.signif, y_position=y.position),
         manual=TRUE, tip_length = 0.,step_increase=0,textsize=4
       )+
      stat_summary(fun.data = "mean_se", geom = "pointrange", show.legend = F,
                   position = position_dodge(.175)) +
      theme_minimal() +
      theme(
        legend.position = 'none',
        plot.title = element_text(hjust = 0.5),
        text = element_text(size = 12),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title.x = element_blank(),
        plot.margin = margin(t = 20, r = 10, b = 10, l = 10, unit = 'pt'),
        axis.line = element_line(colour = "black",
                                 size = 0.3, linetype = "solid"),
        axis.text.x = element_text(size=12, color='black'),
        axis.text.y = element_text(size=12, color='black'),
```

```
        axis.title.y = element_blank(),
        strip.placement = "outside") + ggtitle(i)

    }
  }
}


#lg_legend<-function(a.gplot){
#  tmp <- ggplot_gtable(ggplot_build(a.gplot))
#  leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
#  legend <- tmp$grobs[[leg]]
#  return(legend)}

#leg <- get_legend(p1[[1]])

d <- grid.arrange(grobs = list(p1[[1]],p1[[2]]), left = textGrob("Module score",
rot = 90, vjust = 1),ncol=2)
```



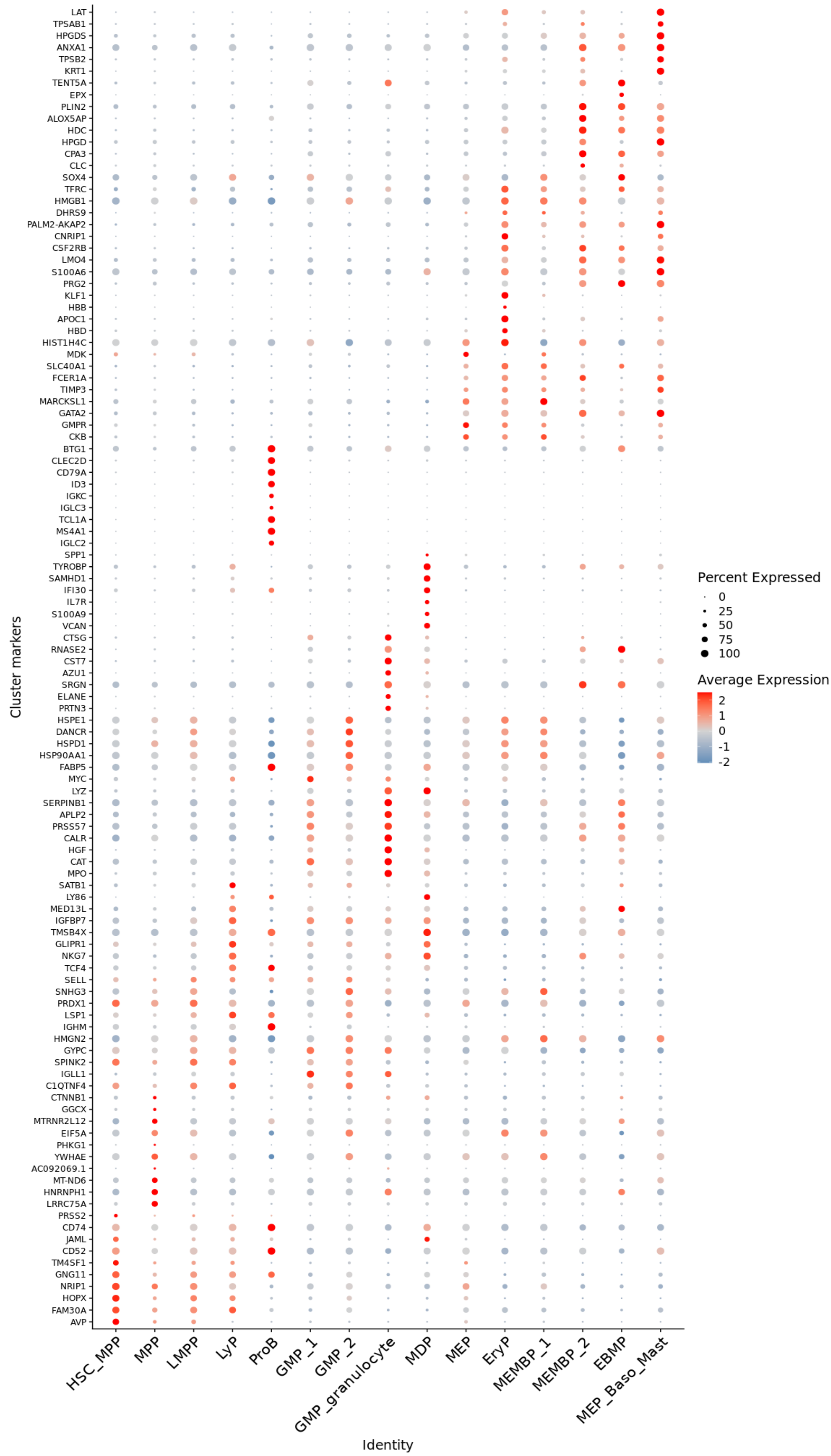

```
#svg(paste(out_dir,'signature_dnarep.svg", sep='/'), width =12, height = 15,
pointsize=12)
#plot(d)
#dev.off()
```

Supplementary Figures

Supplementary Figure 4 a - cluster composition new table and figures

```
Idents(full_obj) <- 'Celltype'
top10 <- full_obj@misc$markers_h$annot_cluster$RNA_snn_h.orig.ident_res.2_refined
%>% group_by(cluster) %>% top_n(n = 10, wt = avg_log2FC)
g <- unique(top10$gene)
d <- DotPlot(object = full_obj, features = g,
             cols = c("white","red"),dot.scale = 3)+ #dot.min
  RotatedAxis()+coord_flip()+xlab('Cluster markers')+
  theme(axis.text.x=element_text(size=16),axis.text.y=element_text(size=9))+
  scale_colour_gradient2(low="steelblue", mid="lightgrey", high="red")+
  theme(
    axis.line = element_line(color = "black"),
    axis.ticks = element_line(color = "black"),
    axis.text = element_text(color = "black")
```

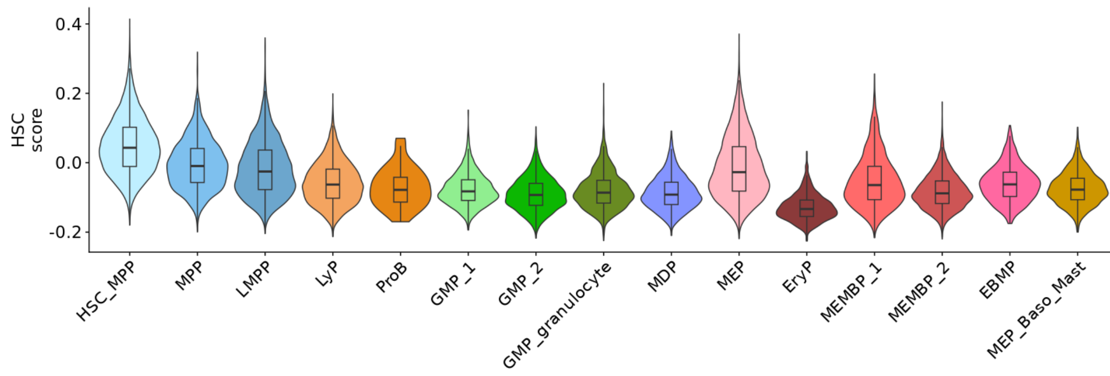
```
  )

#svg(paste(out_dir,"Paper_plots", "Cluster_lineage_markers.svg", sep = "/"), width
=10, height = 15, pointsize=12)
#d
#dev.off()
d
```

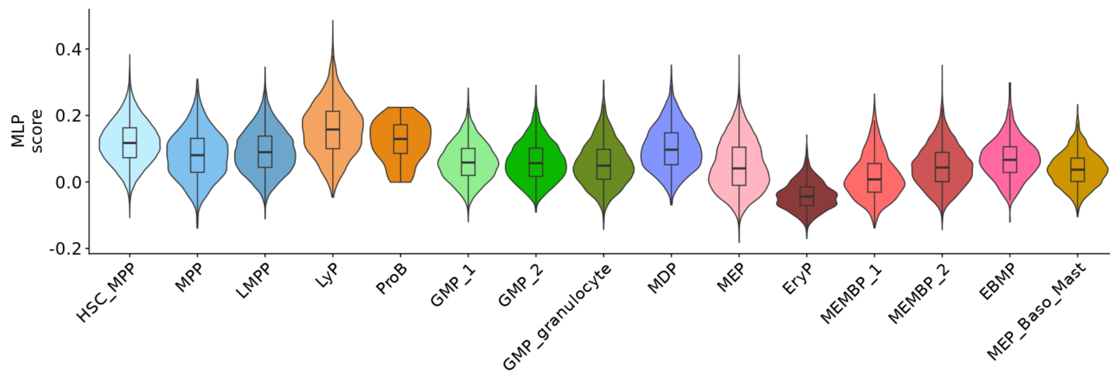

Cluster markers
Identity
HSC_MPP
MPP
LMPP
LyP
ProB
GMP_1
GMP_2
GMP_granulocyte
MDP
MEP
EryP
MEMBP_1
MEMBP_2
EBMP
MEP_Baso_Mast
Percent Expressed
0
25
50
75
100
Average Expression
2
1
0
-1
-2

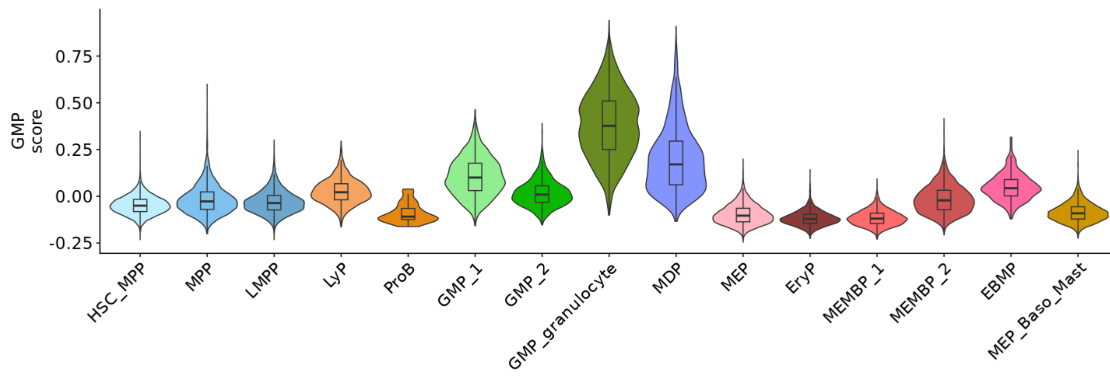
Supplementary figures 4 b - lineages score

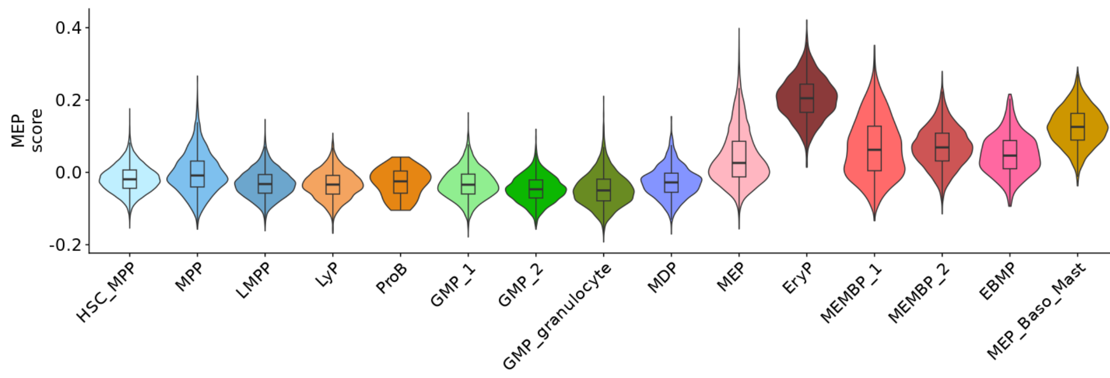
```
## Mende et al, Laurenti lineages
gmt.obj<- read.gmt(paste('reference/GSEA/lineage_hsc_score.gmt',sep='/'))

ll <- unique(gmt.obj$term)

## Calculate module score
# for (l in 1:length(ll)) {
#   group <-ll[l] #levels(gmt.obj$term)[l]
#   nn <- gsub('\\)','',gsub('\\(','',gsub(' ','_',group)))
#   nn <- gsub('-','_',nn)
#   glist <- gmt.obj[gmt.obj$term == group,'gene']
#   gmt_features <- list(glist)
#   #ll <- length(x = which(x = gmt_features %in% rownames(x =
full_obj@assays$RNA)))
#   #if (ll != 0){
#   full_obj <- AddModuleScore(
#     object = full_obj,
#     features = gmt_features, ##
#     #nbin = 24,
#     #ctrl = length(glist), # 100, only for the metabolic pathways
#     #seed = 1,
#     name = nn,
#     #search = TRUE
#
#   )}

#saveRDS(full_obj, paste(out_dir, 'Full_final_v7.rds', sep = '/'))

## GMP score
Idents(full_obj) <- 'Celltype'
l <- unique(gmt.obj$term)
p <- list()
for (t in l) {
  n <- gsub('[.]', '\n',t)
  p[[t]] <- VlnPlot(object = full_obj, features = paste0(t, "1"), #group.by =
c("CSC_clusters"),
                    pt.size = 0,sort = F,col = mycols)+
    geom_boxplot(width=0.2,outlier.shape = NA)+ ylab(n)+
    theme(legend.position="none",title = element_blank(),
          axis.title.x = element_blank(),
          axis.text=element_text(size=16),
          axis.line = element_line(colour = "black", size=0.5),
          axis.title.y = element_text(size=16),
          axis.text.x = element_text(hjust=1,size=16),
          axis.text.y = element_text(size = 16),
          panel.grid = element_blank(),
          panel.border = element_blank(),
          text = element_text( size= 16))+ggtitle('')
```

```
}
## get legend just from one
lg_legend<-function(a.gplot){
  tmp <- ggplot_gtable(ggplot_build(a.gplot))
  leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
  legend <- tmp$grobs[[leg]]
  return(legend)}

legend <- VlnPlot(object = full_obj, features = paste0(t, "1"), #group.by =
c("CSC_clusters"),
                  pt.size = 0,sort = TRUE, col = mycols)+
  geom_boxplot(width=0.2,outlier.shape = NA)

mylegend<- lg_legend(legend)


#svg(paste(out_dir,"Paper_plots", "Lineage_scores_annotated_nonsorted.svg",
sep='/'), width =8, height = 15, pointsize=12)
#grid.arrange(p$HSC.score, p$MLP.score, p$GMP.score,p$MEP.score, #mylegend,
#             nrow = 4,top = textGrob("",gp=gpar(fontsize=20,font=3)))
#dev.off()

grid.arrange(p$HSC.score, p$MLP.score, p$GMP.score,p$MEP.score, #mylegend,
             nrow = 4,top = textGrob("",gp=gpar(fontsize=20,font=3)))
```


HSC
score
0.4
0.2
0.0
-0.2
HSC_MPP
MPP
LMPP
LyP
ProB
GMP_1
GMP_2
GMP_granulocyte
MDP
MEP
EryP
MEMBP_1
MEMBP_2
EBMP
MEP_Baso_Mast


MLP
score
0.4
0.2
0.0
-0.2
HSC_MPP
MPP
LMPP
LyP
ProB
GMP_1
GMP_2
GMP_granulocyte
MDP
MEP
EryP
MEMBP_1
MEMBP_2
EBMP
MEP_Baso_Mast

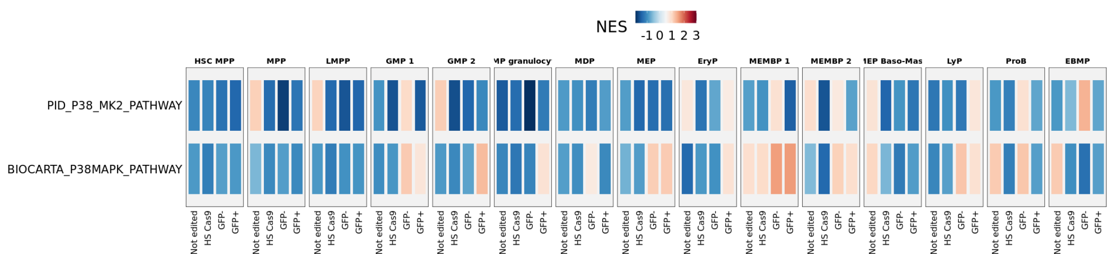

GMP
score
0.75
0.50
0.25
0.00
-0.25
HSC_MPP
MPP
LMPP
LyP
ProB
GMP_1
GMP_2
GMP_granulocyte
MDP
MEP
EryP
MEMBP_1
MEMBP_2
EBMP
MEP_Baso_Mast


MEP
score
0.4
0.2
0.0
-0.2
HSC_MPP
MPP
LMPP
LyP
ProB
GMP_1
GMP_2
GMP_granulocyte
MDP
MEP
EryP
MEMBP_1
MEMBP_2
EBMP
MEP_Baso_Mast

Supplementary figures 4 c - gsea plot

```
hallmark.subset <- NULL
hallmark.subset <-
read.table('Full/Paper_plots/tables/GSEA_results_p38_sig_sel.txt')

min0 <- min(hallmark.subset[!is.na(hallmark.subset$NES),'NES'])
max0 <- max(hallmark.subset[!is.na(hallmark.subset$NES),'NES'])

hallmark.subset$stars=stars.pval(hallmark.subset$pvalue)

hallmark.subset$ID <- str_wrap(hallmark.subset$ID ,width = 45)
hallmark.subset$Name <-  as.character(hallmark.subset$ID)
hallmark.subset$celltype <- gsub('_', ' ', hallmark.subset$celltype)
hallmark.subset$celltype <- ifelse(hallmark.subset$celltype == 'MEP Baso Mast',
'MEP Baso-Mast', hallmark.subset$celltype)

hallmark.subset$celltype <- factor(hallmark.subset$celltype, levels =
                                     c("HSC MPP","MPP","LMPP","GMP 1",
                                       "GMP 2","GMP granulocyte",
"MDP","MEP","EryP","MEMBP 1",
                                       "MEMBP 2", "MEP Baso-
Mast","LyP","ProB","EBMP" ))

hallmark.order <- hallmark.subset%>% group_by(ID) %>% summarise(Pos = sum(NES))
hallmark.order.terms <- hallmark.order[order(hallmark.order$Pos, decreasing =
TRUE), "ID", drop = FALSE]
hallmark.subset$ID <- factor(hallmark.subset$ID, levels = hallmark.order.terms$ID)


hallmark.subset$Dataset <- factor(hallmark.subset$Dataset,
levels=c("UT_p38i_pop_vs_UT_DMSO_pop",

"HS_p38i_pop_vs_HS_DMSO_pop",

"GE_p38i_pop_GFPneg_vs_GE_DMSO_pop_GFPneg",

"GE_p38i_pop_GFPpos_vs_GE_DMSO_pop_GFPpos") )

hallmark.subset$Dataset <- ifelse(hallmark.subset$Dataset ==
"UT_p38i_pop_vs_UT_DMSO_pop", 'Not edited',
                           ifelse(hallmark.subset$Dataset ==
"HS_p38i_pop_vs_HS_DMSO_pop", 'HS Cas9',
                           ifelse(hallmark.subset$Dataset ==
"GE_p38i_pop_GFPneg_vs_GE_DMSO_pop_GFPneg", "GFP-",
                           ifelse(hallmark.subset$Dataset ==
"GE_p38i_pop_GFPpos_vs_GE_DMSO_pop_GFPpos", "GFP+", 'none' ))))

hallmark.subset$Dataset <- factor(hallmark.subset$Dataset, levels=c('Not edited',
```

```
                                                                        'HS Cas9',
                                                                        "GFP-",
                                                                        "GFP+"))


p_p38 <- ggplot(data = hallmark.subset, mapping = aes(x = Dataset, y =
reorder(Name, -NES))) +
  geom_tile(aes(fill=NES, group = celltype), width=0.80, height=0.80)+
  facet_grid(Category ~ celltype  , scales = "free_y", space = "free", )+
  theme(text= element_text(size=22,color='black'),
        axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1,
color='black', size =12),
        axis.text.y = element_text(color='black', size =16),
        plot.title = element_text(hjust = 0), plot.subtitle = element_text(hjust =
0),
        axis.ticks = element_blank(), axis.text = element_text(size = 8),
axis.title = element_blank(),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_rect(fill = "gray95", color='black'),
        strip.background = element_rect(colour="white",
                                        fill="white",size=0.5),
        strip.text = element_text(color='black',face = 'bold', size = 11),
        legend.position = 'top', legend.direction = 'horizontal') +
  scale_fill_gradientn(colours =
colorRampPalette(rev(brewer.pal(11,"RdBu")))(100),
                       limits = c(min0,3),
                       na.value = "grey")


#wdir='Full/DGE_analysis'
#svg(paste(wdir,"HM_categories_v3_P38_pid.svg",sep = '/'),width=25,
#    height=3,   pointsize=12)
#p
#dev.off()
p_p38
```



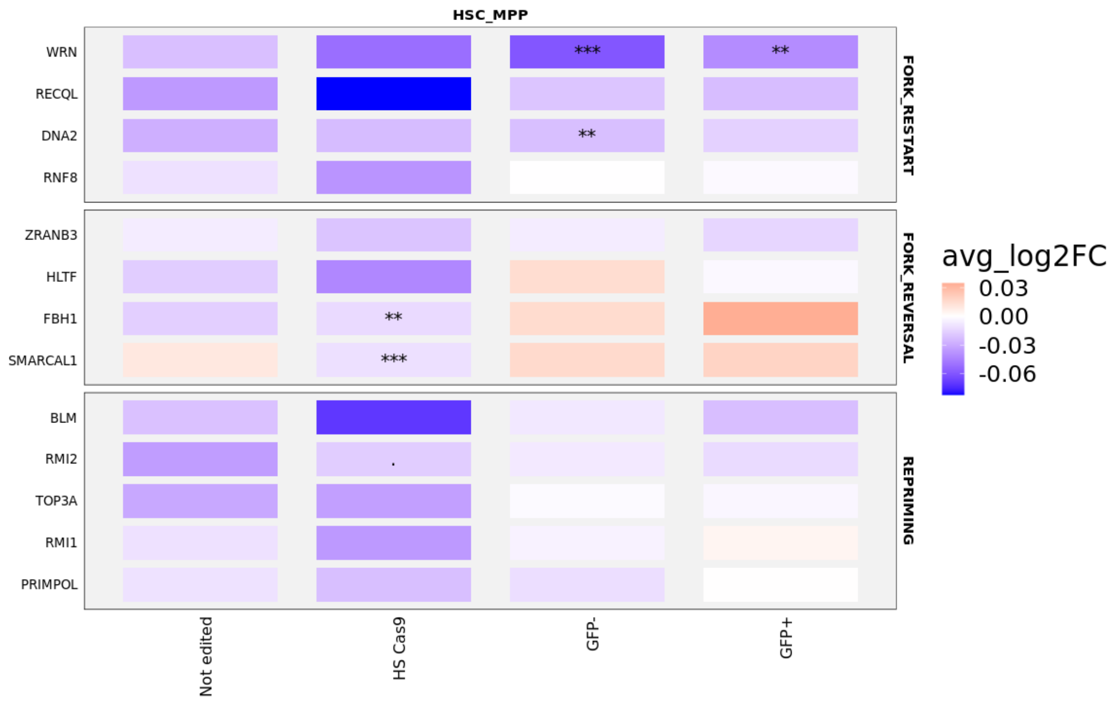
Supplementary figures 4 e - Lopes et al genes

```
################################ plot gene for each cluster and for each condition
c <- list("UT_p38i_pop_vs_UT_DMSO_pop" = c("UT_p38i_pop","UT_DMSO_pop"),
```

```
          "HS_p38i_pop_vs_HS_DMSO_pop" = c( "HS_p38i_pop","HS_DMSO_pop"),
          "GE_DMSO_pop_GFP+_vs_GE_DMSO_pop_GFP-" =
c("GE_DMSO_pop_GFP+","GE_DMSO_pop_GFP-"), ## h3 vs h3
          "GE_p38i_pop_GFP+_vs_GE_DMSO_pop_GFP+" =
c("GE_p38i_pop_GFP+","GE_DMSO_pop_GFP+"), ## h4 vs h3
          "GE_p38i_pop_GFP-_vs_GE_DMSO_pop_GFP-" = c("GE_p38i_pop_GFP-
","GE_DMSO_pop_GFP-"),
          "GE_DMSO_pop_GFP+_vs_UT_DMSO_pop" = c("GE_DMSO_pop_GFP+","UT_DMSO_pop"),
## h3 vs h3
          "GE_DMSO_pop_GFP-_vs_UT_DMSO_pop" = c("GE_DMSO_pop_GFP-","UT_DMSO_pop"),
          "GE_p38i_pop_GFP+_vs_UT_DMSO_pop" = c("GE_p38i_pop_GFP+","UT_DMSO_pop"),
          "GE_p38i_pop_GFP-_vs_UT_DMSO_pop" = c("GE_p38i_pop_GFP-","UT_DMSO_pop"))
## h3 vs h3

## dge analysis tables
marker_dir <- 'Full/DGE_analysis/DGE_alivecd90_together/tables'


Idents(full_obj) <- full_obj$Celltype

x <- unique(full_obj$Celltype)
x <- c('MEMBP_2', 'LMPP', 'LyP','HSC_MPP','EryP',
       'MEP', 'GMP_2', 'MEMBP_1','MPP','GMP_granulocyte', 'MEP_Baso_Mast',
       'GMP_1','MDP', 'EBMP',  'ProB' )
xnew <- paste0(as.character(x),'_')

cnew <- paste0('_',as.character(names(c)))
cnew <- gsub('\\+', 'pos',cnew)
cnew <- gsub('\\-', 'neg',cnew)


comparisons <- expand.grid(x,names(c))
comp <- paste0(comparisons$Var1, '_', comparisons$Var2)
markers_all <- data.frame()
for (compx in comp) {  #for each comparisons withina cluster
  #cat(compx, '\n')
  ### get name comparison and celltype separated
  compx_original <- compx
  compx <- gsub('\\+', 'pos',compx)
  compx <- gsub('\\-', 'neg',compx)
  celltype <- unique(str_replace(compx,cnew,''))

  cp <- celltype[order(nchar(celltype), celltype)][1] ## cell type

  nx <- unique(str_replace_all(compx,as.character(xnew),''))
  n <- nx[order(nchar(nx), nx)][1]  ## comparisons
  f <- paste(marker_dir, paste0(compx_original, '.txt'),sep = '/')

  if (file.exists(f)) {
    markers <- read.table(f)
```

```
    markers$Celltype <- cp
    markers$Contrast <- n
    markers$Comparisons <- compx
    markers$gene <- rownames(markers)
    rownames(markers) <- NULL
  }
  if (nrow(markers_all) == 0) {
    markers_all <- markers
  } else {
    markers_all <- rbind(markers_all,markers)
  }

}
rownames(markers_all) <- NULL

markers_all <- markers_all[order(markers_all$avg_log2FC),]


genes <- list('FORK_REVERSAL'=c('FBH1','SMARCAL1','ZRANB3','HLTF'),
              'FORK_RESTART' = c('RECQL','WRN','RNF8','DNA2'),
              'REPRIMING' = c('BLM','RMI1','TOP3A','RMI2','PRIMPOL'))

markers_all_genes <- markers_all
hsc_genes <- markers_all_genes[markers_all_genes$gene %in% unname(unlist(genes)) &
markers_all_genes$Celltype == 'HSC_MPP',]


#print(hsc_genes[order(hsc_genes$avg_log2FC, decreasing = FALSE), ]   )

markers_all_subset <- hsc_genes

markers_all_subset$pvalue=stars.pval(markers_all_subset$p_val_adj)

markers_all_subset$Category <- ifelse(markers_all_subset$gene %in%
genes$FORK_REVERSAL, 'FORK_REVERSAL',
                                      ifelse(markers_all_subset$gene %in%
genes$FORK_RESTART, 'FORK_RESTART',
                                             'REPRIMING'))

min0 <- min(markers_all_subset$avg_log2FC)
max0 <- max(markers_all_subset$avg_log2FC)
markers_all_subset <- markers_all_subset[markers_all_subset$Contrast %in%
c("UT_p38i_pop_vs_UT_DMSO_pop", "GE_p38i_pop_GFPpos_vs_GE_DMSO_pop_GFPpos",

"GE_p38i_pop_GFPneg_vs_GE_DMSO_pop_GFPneg","HS_p38i_pop_vs_HS_DMSO_pop" ),]

markers_all_subset$treatment <- ifelse(markers_all_subset$Contrast ==
"UT_p38i_pop_vs_UT_DMSO_pop", 'Not edited',
                                ifelse(markers_all_subset$Contrast ==
"HS_p38i_pop_vs_HS_DMSO_pop", 'HS Cas9',
```

```
                                ifelse(markers_all_subset$Contrast ==
"GE_p38i_pop_GFPneg_vs_GE_DMSO_pop_GFPneg", 'GFP-',
                                ifelse(markers_all_subset$Contrast ==
"GE_p38i_pop_GFPpos_vs_GE_DMSO_pop_GFPpos",'GFP+','nd'))))

markers_all_subset$treatment <- factor(markers_all_subset$treatment, levels=c('Not
edited','HS Cas9',"GFP-","GFP+") )
markers_all_subset$Celltype <- ifelse(markers_all_subset$Celltype == 'CMP',
'MPP',markers_all_subset$Celltype )
markers_all_subset$Celltype <- factor( markers_all_subset$Celltype,levels =
levels(full_obj$Celltype))

hsc_plot <- ggplot(markers_all_subset, aes(y =  reorder(gene, -avg_log2FC), x
=treatment, fill=avg_log2FC)) +
  geom_tile(aes(fill=avg_log2FC, group = Celltype), width=0.80, height=0.80)+
  facet_grid(Category ~ Celltype  , scales = "free_y", space = "free", )+
  theme(text= element_text(size=22,color='black'),
        axis.text.x = element_text(angle = 90, vjust = 1, hjust =1, color='black',
size =12),
        axis.text.y = element_text(color='black', size =10),
        plot.title = element_text(hjust = 0), plot.subtitle = element_text(hjust =
0),
        axis.ticks = element_blank(), axis.text = element_text(size = 8),
axis.title = element_blank(),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_rect(fill = "gray95", color='black'),
        strip.background = element_rect(colour="white",
                                        fill="white",size=0.5),
        strip.text = element_text(color='black',face = 'bold', size = 11)) +
  geom_text(aes(label = pvalue),size =5) +
  scale_fill_gradientn(colours = c('blue', 'white', 'red'),
                       na.value = "grey",
                       rescaler = ~ scales::rescale_mid(.x, mid = 0))+
  ggtitle('p38i vs DMSO')

#svg(paste('Full/DGE_analysis',"DGE_Lopes_genes.svg",sep = '/'),width=6,
#    height=12,   pointsize=12)
#hsc_plot
#dev.off()
hsc_plot
```



p38i vs DMSO
HSC_MPP
WRN
RECQL
DNA2
RNF8
ZRANB3
HLTF
FBH1
SMARCAL1
BLM
RMI2
TOP3A
RMI1
PRIMPOL
***
**
**
**
***
.
FORK_RESTART
FORK_REVERSAL
REPRIMING
Not edited
HS Cas9
GFP-
GFP+
avg_log2FC
0.03
0.00
-0.03
-0.06