B/plasma figures

inputs

In [1]:

```
source('jupyterFunctions_perCellType.R')
```

In [2]:

```
CT <- 'Bplasma'
CT_label <- 'B/plasma'
data_prefix <- paste(sep='','../data/',CT,'/',CT)
ATAC_meta <- readRDS(paste(sep='',data_prefix,'_ATAC_meta.rds'))
chosenPeaks <- readRDS(paste(sep='',data_prefix,'_chosenPeaks.rds'))
snATAC_pxc_norm <- readRDS(paste(sep='',data_prefix,'_snATAC_pxc_norm.rds'))
snRNA_gxc_norm <- readRDS(paste(sep='',data_prefix,'_snRNA_gxc_norm.rds'))
snATAC_pxCT_norm <- readRDS(paste(sep='',data_prefix,'_snATAC_pxCT_norm.rds'))
snRNA_gxCT_norm <- readRDS(paste(sep='',data_prefix,'_snRNA_gxCT_norm.rds'))
chromVARz_mat <- readRDS(paste(sep='',data_prefix,'_ArchR_chromVARz_JASPAR2020.rds'))
ArchR_padj <- readRDS(paste(sep='',data_prefix,'_ArchR_padj_JASPAR2020.rds'))
CITE_meta <- readRDS(paste(sep='',data_prefix,'_CITE_meta.rds'))
class_state_df <- readRDS(paste(sep='',data_prefix,'_class_state_df.rds'))
LDA_res <- readRDS(paste(sep='',data_prefix,'_LDA_stats.rds'))
```

In [3]:

```
ATAC_colors <- readRDS('../data/misc/ATAC_class_colors.rds')
CITE_colors <- readRDS('../data/misc/CITE_state_colors.rds')
ATAC_CITE_conv_df <- readRDS('../data/misc/ATAC_CITE_sample_conversion.rds')
```

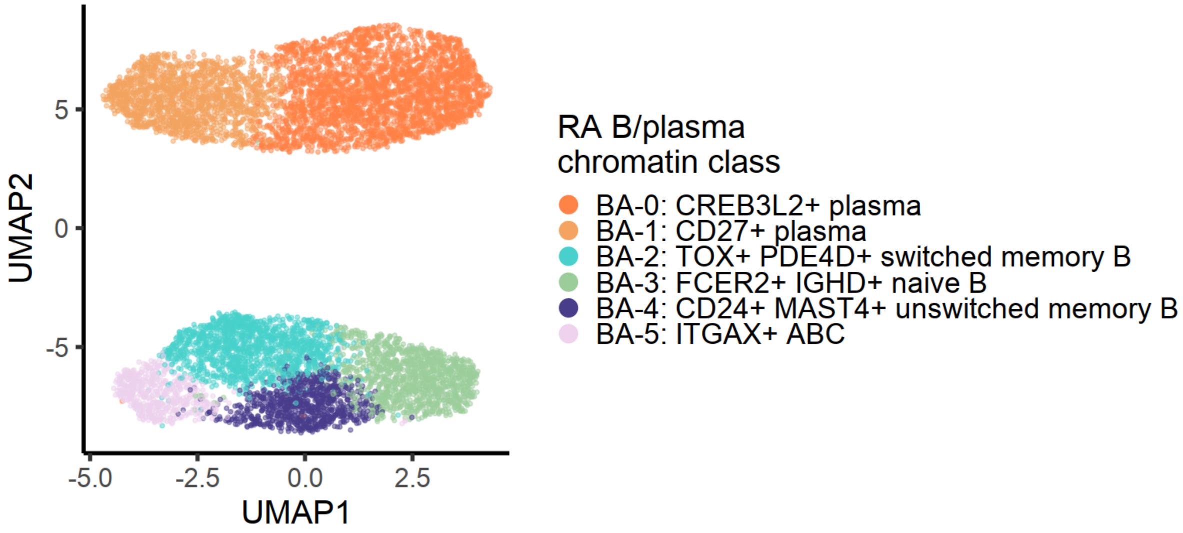
In [22]:

```
save_dir <- NA #'../output/' #or NA if don't want to save
```

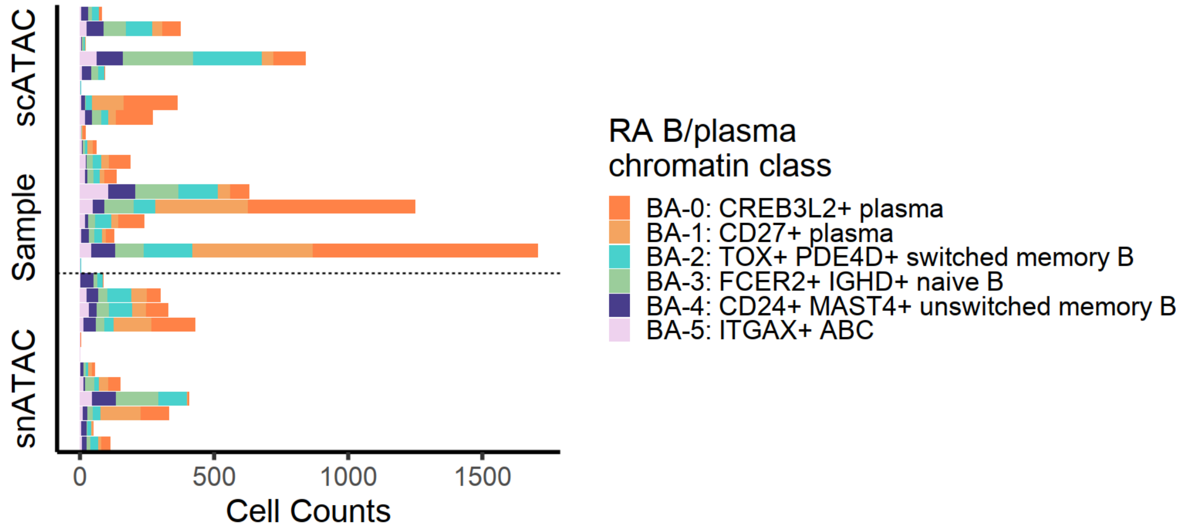
ATAC classes

In [5]:

```
#Fig 5a
options(repr.plot.height=6,repr.plot.width=13)
g <- ggplot(ATAC_meta,aes_string(x='UMAP1',y='UMAP2',color='cluster_name')) + geom_point(size=1,alpha=0.5) +
        theme_classic(base_size=25) + scale_color_manual(values=ATAC_colors) +
        labs(color=paste(sep='','RA ',CT_label,'\nchromatin class')) +
        theme(legend.text=element_text(size=22)) +
        guides(colour = guide_legend(override.aes = list(size=6,alpha=1)))
print(g)

if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_ATAC_class_UMAP.png'),
                            plot=g,units='in',height=6,width=13,dpi=600)
```



In [6]:
```
#Fig S5a
options(repr.plot.height=6,repr.plot.width=13)
g <- cellCount_bySample_barPlot(ATAC_meta,'sample','cluster_name',paste(sep='','RA ',CT_label,'\nchromatin class'),
                                ATAC_colors)
print(g)

if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_ATAC_class_cellCount.png'),
                            plot=g,units='in',height=6,width=13,dpi=600)
```

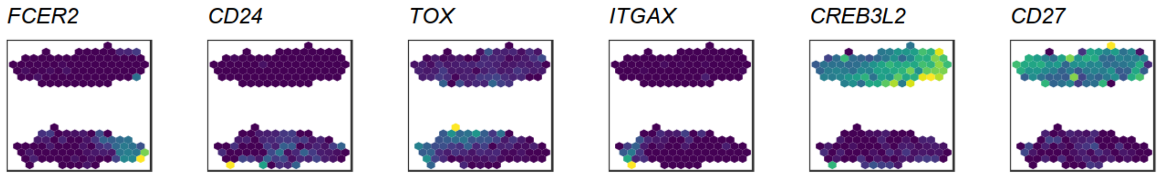


ATAC cluster markers

In [7]:
```
chosenGenes <- names(chosenPeaks)
chosenPeaks <- chosenPeaks[!is.na(chosenPeaks)] #NA means no peak in gene's promoter
```

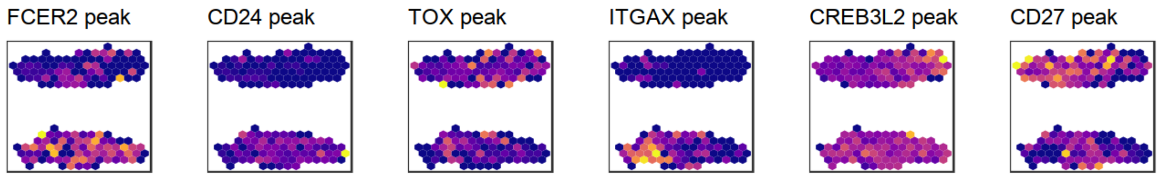
In [8]:

```
#Fig 5b bottom

genes_forUMAPs <- c('FCER2','CD24','TOX','ITGAX','CREB3L2','CD27')
if(!all(genes_forUMAPs %in% names(chosenPeaks))) stop('Genes for UMAP not in chosen genes')

multiome_cells <- rownames(ATAC_meta[which(ATAC_meta$assay=='snATAC'),])

options(repr.plot.height=2,repr.plot.width=11)
g <- plot_markerPeaks_norm_hex_v2(ATAC_meta[multiome_cells,],snRNA_gxc_norm[genes_forUMAPs,multiome_cells],'UMAP1','UMAP2',
                                  plot_genes=genes_forUMAPs,plotCol=length(genes_forUMAPs),
                                  titleSize=14,hex_bins=16,cutCap=0)
grid.draw(g)
if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_markerGene_UMAP.png'),
                            plot=g,units='in',height=2,width=11,dpi=600)
```



In [9]:

```
#Fig 5b top

toPlot <- snATAC_pxc_norm[unname(chosenPeaks[genes_forUMAPs]),multiome_cells]
rownames(toPlot) <- paste(sep='',names(chosenPeaks[genes_forUMAPs]),' peak')

options(repr.plot.height=2,repr.plot.width=11)
g <- plot_markerPeaks_norm_hex_v2(ATAC_meta[multiome_cells,],toPlot,'UMAP1','UMAP2',
                                  plot_genes=rownames(toPlot),plotCol=nrow(toPlot),titleSize=14,hex_bins=16,cutCap=0,
                                  titleFace='plain',colorOpt='plasma')
grid.draw(g)
if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_markerPeak_UMAP.png'),
                            plot=g,units='in',height=2,width=11,dpi=600)
```



In [10]:

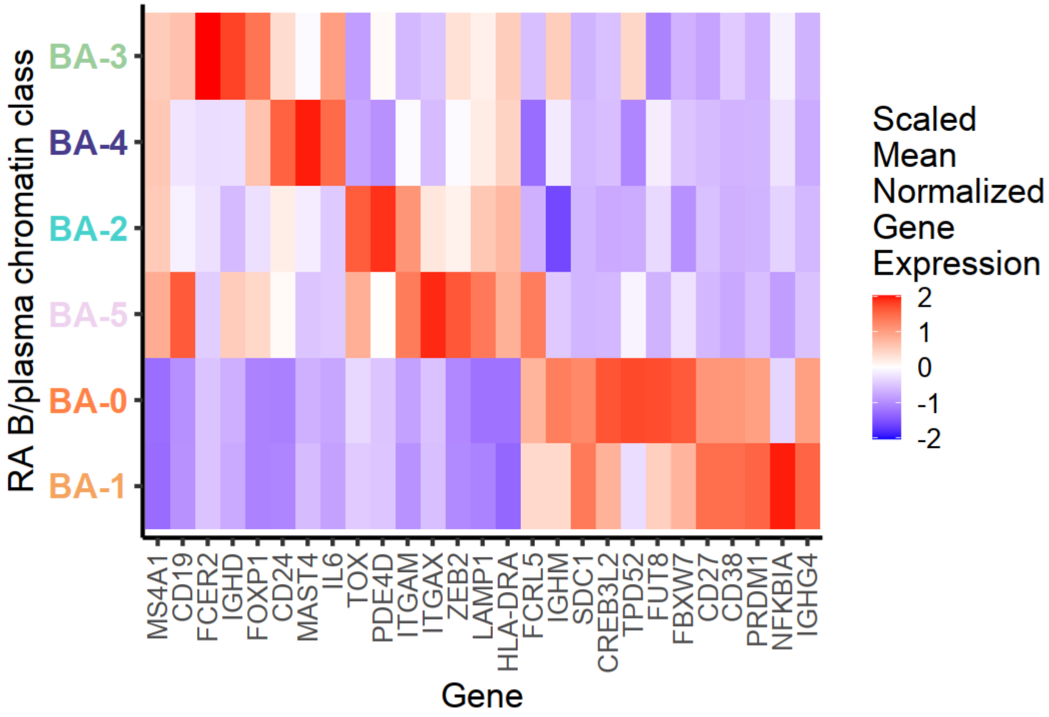
```
class_order <- c('BA-3','BA-4','BA-2','BA-5','BA-0','BA-1')
all(class_order %in% ATAC_meta$cluster_abbr)
```

TRUE

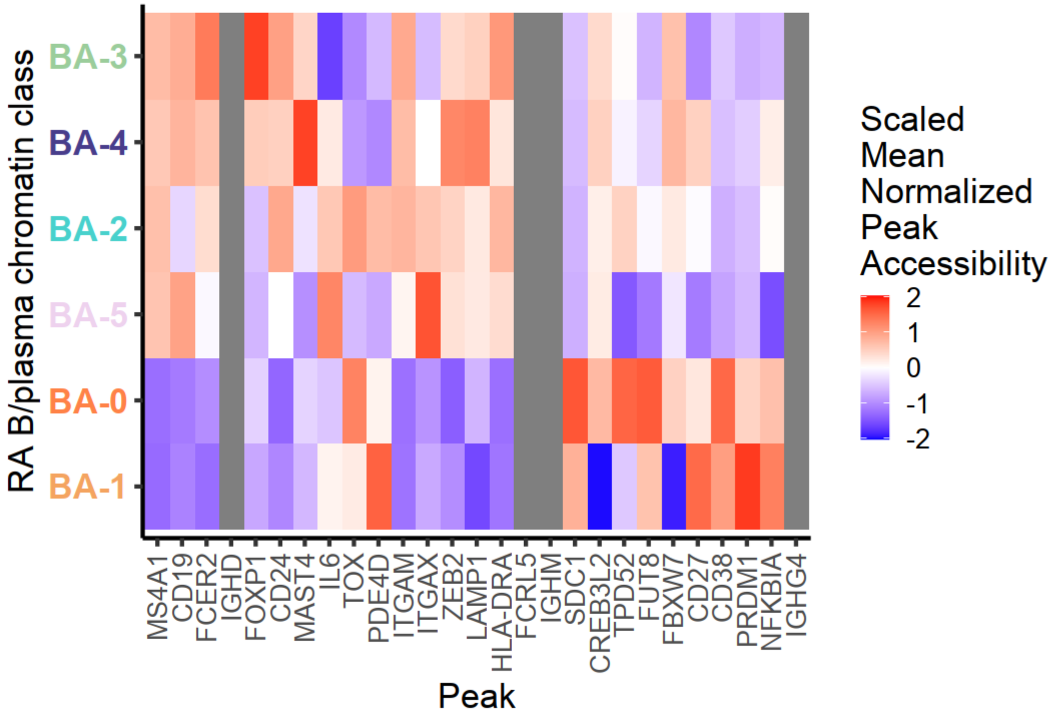
In [11]:

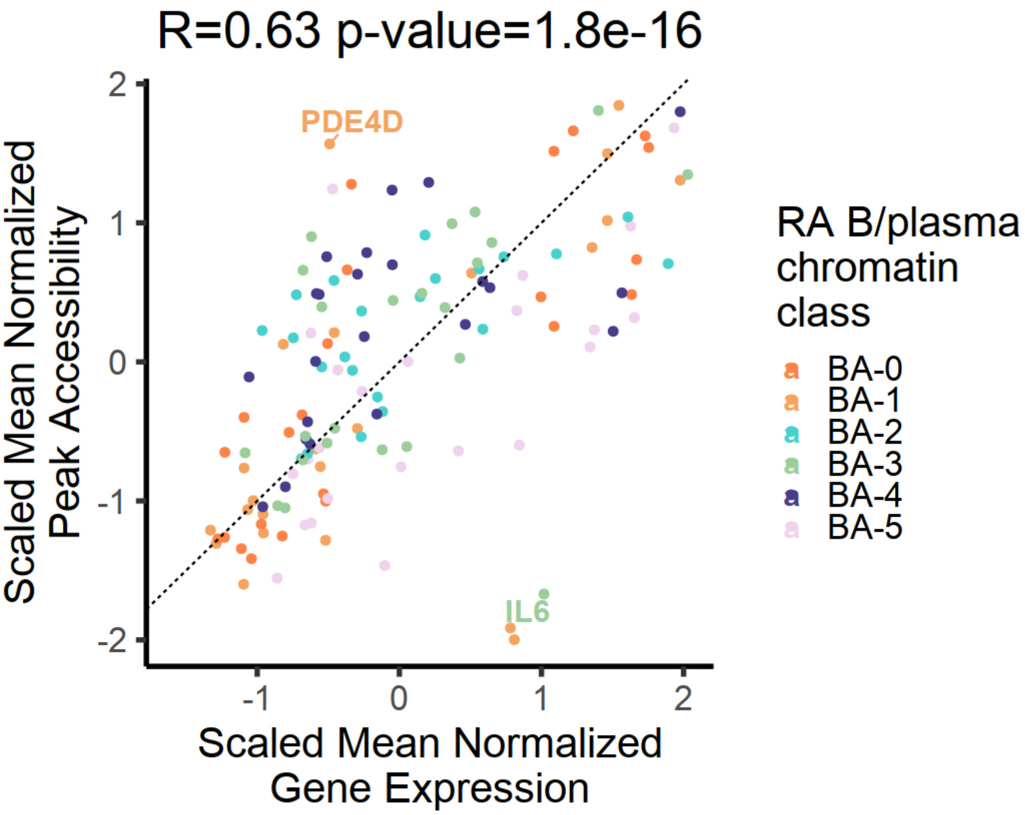
```
#Fig S5b

res <- scaleFeat_forHeatmap(chosenGenes,class_order,chosenPeaks,snRNA_gxCT_norm,snATAC_pxCT_norm)
snRNA_gxCT_norm_subset_scaled <- res$gxCT_norm_subset_scaled
snATAC_pxCT_norm_subset_scaled <- res$pxCT_norm_subset_scaled
fxCT_norm_subset_scaled <- res$fxCT_norm_subset_scaled

scale_lim <- max(abs(snRNA_gxCT_norm_subset_scaled),abs(snATAC_pxCT_norm_subset_scaled),na.rm=TRUE)

options(repr.plot.height=7,repr.plot.width=9)
g <- pseudobulk_scaled_heatmap(snRNA_gxCT_norm_subset_scaled,'Gene',paste('RA',CT_label,'chromatin class'),
                               'Scaled\nMean\nNormalized\nGene\nExpression',
                               plotTit=paste('Scaled Mean Normalized Gene Expression\nof multiome cells by RA',
                                             CT_label,'chromatin classes'),
                               scale_lim=scale_lim,clustColors=ATAC_colors)
print(g)
if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_markerGene_heatmap.png'),
                            plot=g,units='in',height=7,width=9,dpi=600)

g <- pseudobulk_scaled_heatmap(snATAC_pxCT_norm_subset_scaled,'Peak',paste('RA',CT_label,'chromatin class'),
                               'Scaled\nMean\nNormalized\nPeak\nAccessibility',
                               plotTit=paste('Scaled Mean Normalized Peak Accessibility\nof multiome cells by RA',
                                             CT_label,'chromatin classes'),
                               scale_lim=scale_lim,clustColors=ATAC_colors)
print(g)
if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_markerPeak_heatmap.png'),
                            plot=g,units='in',height=7,width=9,dpi=600)

pearR <- cor.test(fxCT_norm_subset_scaled$gene_norm_scale,fxCT_norm_subset_scaled$peak_norm_scale,
               method='pearson')

fxCT_norm_subset_scaled$label <- ''
fxCT_norm_subset_scaled[which(fxCT_norm_subset_scaled$gene=='PDE4D' & fxCT_norm_subset_scaled$cluster_abbr=='BA-1'),
                        'label'] <- 'PDE4D'
fxCT_norm_subset_scaled[which(fxCT_norm_subset_scaled$gene=='IL6' & fxCT_norm_subset_scaled$cluster_abbr=='BA-3'),
                        'label'] <- 'IL6'

g <- ggplot(fxCT_norm_subset_scaled,
            aes_string(x='gene_norm_scale',y='peak_norm_scale',color='cl
uster_abbr',label='label')) +
        geom_point(size=2) + theme_classic(base_size=25) + scale_color_m
anual(values=ATAC_colors) +
        labs(x='Scaled Mean Normalized\nGene Expression',
             y='Scaled Mean Normalized\nPeak Accessibility',
             color=paste(sep='','RA ',CT_label,'\nchromatin\nclass')) +
        geom_abline(slope=1,intercept=0,linetype='dashed') +
        ggtitle(paste(sep='','R=',round(pearR$estimate,2),' p-value=',si
gnif(pearR$p.value,3))) +
        theme(plot.title=element_text(hjust = 0.5)) + geom_text_repel(bo
x.padding = 0.5,size=6.5,fontface='bold',seed=0)
suppressWarnings(print(g)) #points excluded if peak does not exist
if(!is.na(save_dir)) suppressWarnings(ggsave(file=paste(sep='',save_dir,
CT,'_markerGenePeak_scatterplot.png'),
                                             plot=g,units='in',height=7,
width=9,dpi=600))
```

Scaled Mean Normalized Gene Expression
of multiome cells by RA B/plasma chromatin classes



Scaled Mean Normalized Peak Accessibility
of multiome cells by RA B/plasma chromatin classes

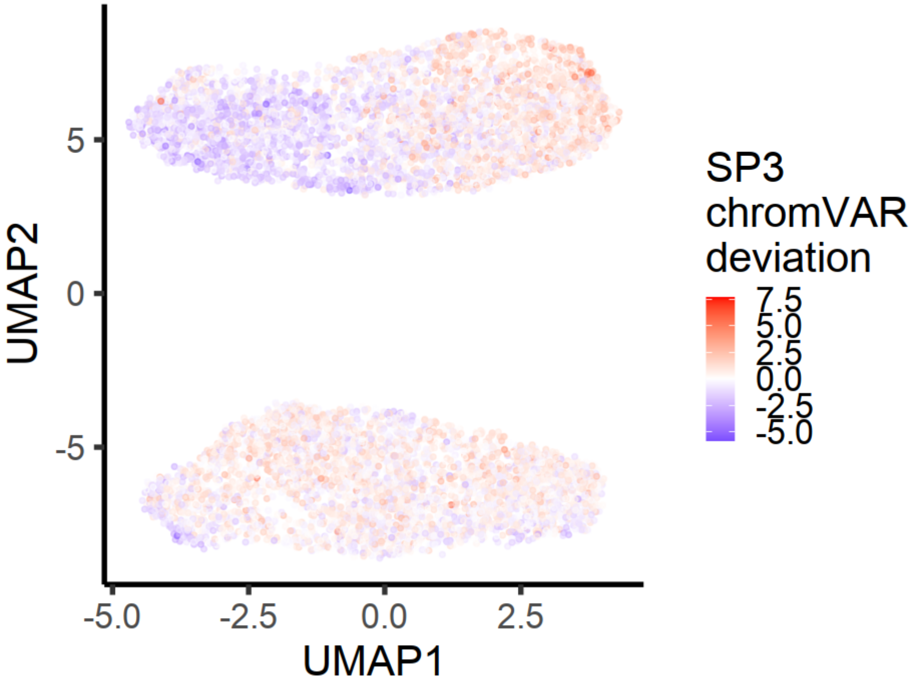



R=0.63 p-value=1.8e-16
PDE4D
IL6
Scaled Mean Normalized
Peak Accessibility
2
1
0
-1
-2
-1
0
1
2
Scaled Mean Normalized
Gene Expression
RA B/plasma
chromatin
class
BA-0
BA-1
BA-2
BA-3
BA-4
BA-5

TFs

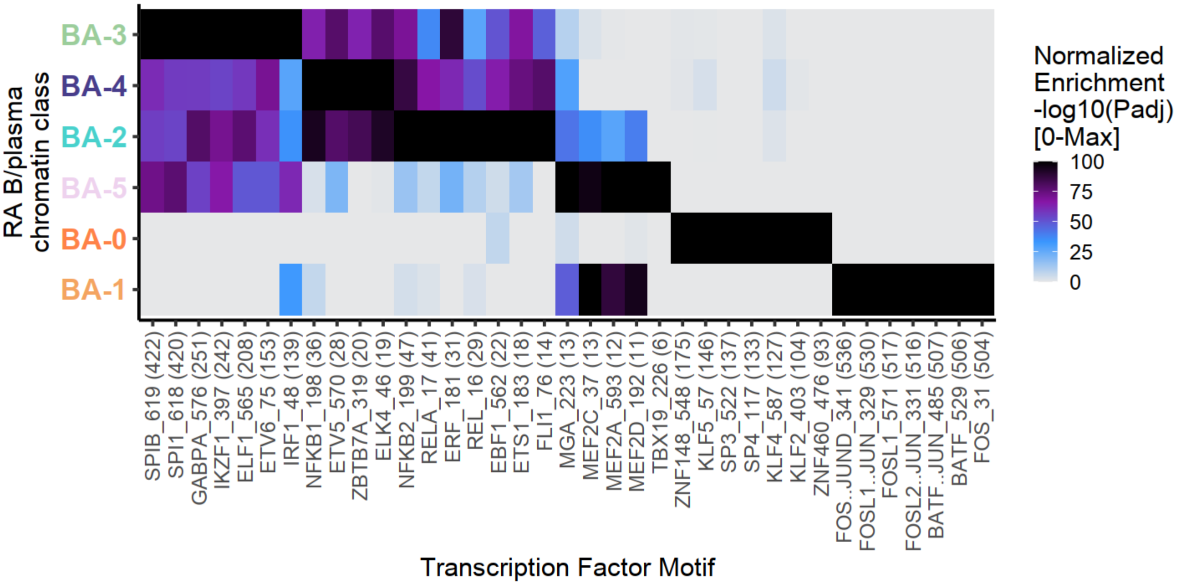
In [12]:

```
#Fig 5c left

#fix cell names
split1 <- str_split_fixed(colnames(chromVARz_mat),'#',2)
new_colnames <- paste(sep='',split1[,1],'_',str_split_fixed(split1[,
2],'-',2)[,1])
#ArchR removed donors with two or fewer cells, so one sample with only t
wo B/plasma cells was removed
cells_kept <- rownames(ATAC_meta[which(rownames(ATAC_meta) %in% new_coln
ames),])
colnames(chromVARz_mat) <- new_colnames

motif_toPlot <- 'SP3_522'
toPlot <- cbind(ATAC_meta[cells_kept,],'motif'=chromVARz_mat[motif_toPlo
t,cells_kept])

options(repr.plot.height=6,repr.plot.width=8)
g <- ggplot(toPlot,aes_string(x='UMAP1',y='UMAP2',color='motif')) + geom
_point(size=1,alpha=0.5) +
        theme_classic(base_size=25) + scale_color_gradient2(low='blue',m
id='white',high='red',midpoint=0) +
        labs(color=paste(sep='',str_split_fixed(motif_toPlot,'_',2)[,
1],'\nchromVAR\ndeviation'))
print(g)

if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_motif_',moti
f_toPlot,'_UMAP.png'),
                            plot=g,units='in',height=6,width=8,dpi=600)
```

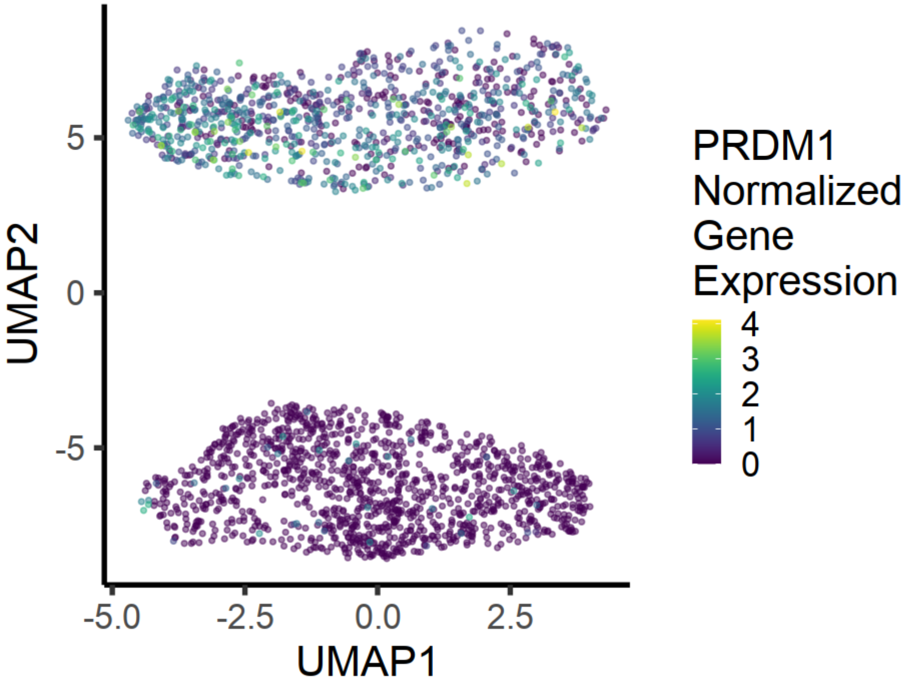


In [13]:

```
#Fig 5c right

#add hyphen back
if(!identical(sort(colnames(ArchR_padj)),sort(colnames(snRNA_gxCT_nor
m))) &
   all(str_detect(colnames(ArchR_padj),'^[a-zA-Z]{2}[0-9]+$'))){
    colnames(ArchR_padj) <- lapply(colnames(ArchR_padj),FUN=function(s)
{paste(sep='',substr(s,1,2),'-',

substr(s,3,nchar(s)))})
}
if(!identical(sort(colnames(ArchR_padj)),
              sort(colnames(snRNA_gxCT_norm)))) stop('mxCT and gxCT matr
ices do not have same CT.')


options(repr.plot.height=6,repr.plot.width=12)
g <- ArchR_topMotifs_KWspin(ArchR_padj,snRNA_gxCT_norm,cOrd=class_order,
cColors=ATAC_colors,
                            minE=5,num_mot=7,minGE=0.05,withinE=0.95,
                            mLab='Transcription Factor Motif',cLab=paste
(sep='','RA ',CT_label,'\nchromatin class'))
print(g)

if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_motif_heatma
p.png'),
                            plot=g,units='in',height=6,width=12,dpi=600)
```



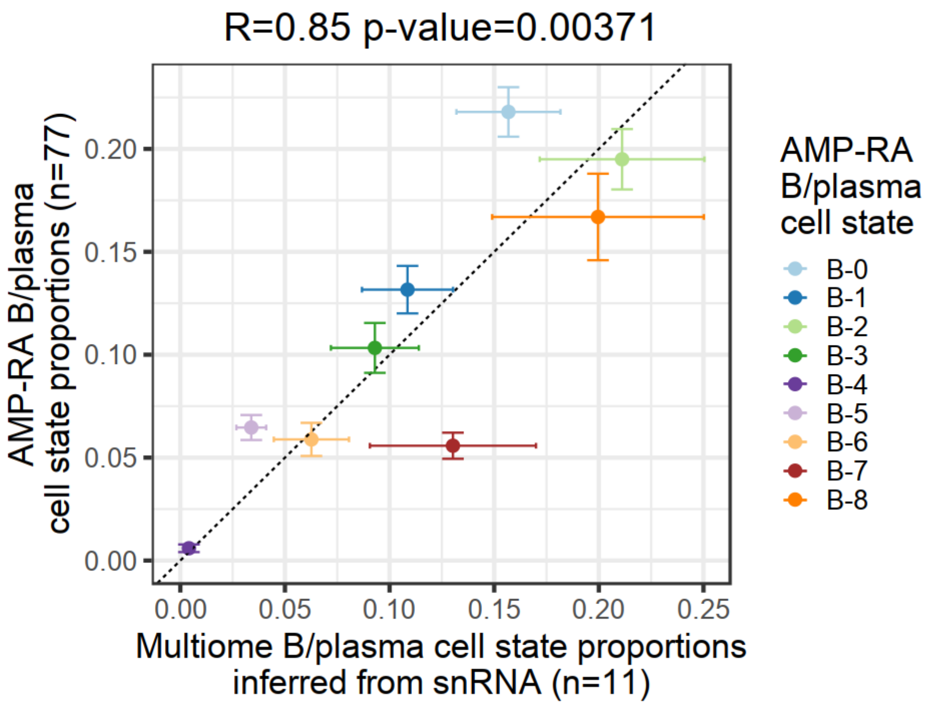
In [14]:
```
#Fig 5d left

gene_toPlot <- 'PRDM1'
toPlot <- cbind(ATAC_meta[multiome_cells,],'gene'=snRNA_gxc_norm[gene_toPlot,multiome_cells])

options(repr.plot.height=6,repr.plot.width=8)
g <- ggplot(toPlot[order(toPlot$gene),],aes_string(x='UMAP1',y='UMAP2',color='gene')) +
                geom_point(size=1,alpha=0.5) +
                theme_classic(base_size=25) + scale_color_viridis(option = 'viridis') +
                labs(color=paste(sep='',gene_toPlot,'\nNormalized\nGene\nExpression'))
print(g)

if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_gene_',gene_toPlot,'_UMAP.png'),
                            plot=g,units='in',height=6,width=8,dpi=600)
```



Transcriptional Cell States

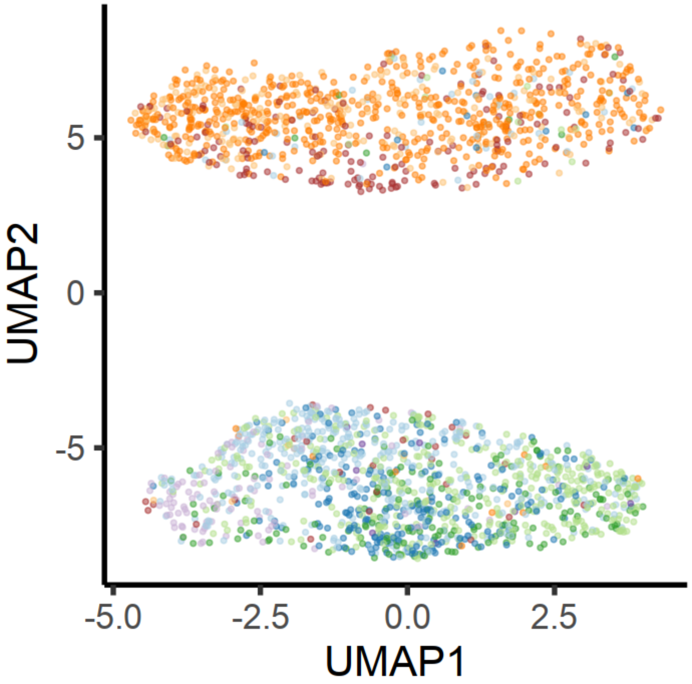
In [15]:

```
#Fig S8d

options(repr.plot.height=6,repr.plot.width=8)
g <- symp_prop_df(ATAC_meta[multiome_cells,],CITE_meta,
                  paste(sep='','Multiome ',CT_label,' cell state proport
ions\ninferred from snRNA (n=',
                        length(unique(ATAC_meta[multiome_cells,'sampl
e'])),')'),
                  paste(sep='','AMP-RA ',CT_label,'\ncell state proporti
ons (n=',
                        length(unique(CITE_meta$sample)),')'),
                  paste(sep='','AMP-RA\n',CT_label,'\ncell state'),clust
Colors=CITE_colors)
print(g)

if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_ATAC_CITE_st
ate_prop.png'),
                            plot=g,units='in',height=6,width=8,dpi=600)
```



In [16]:

```
#Fig S8g left
options(repr.plot.height=6,repr.plot.width=6)
g <- ggplot(ATAC_meta[multiome_cells,],aes_string(x='UMAP1',y='UMAP2',color='CITE')) +
        geom_point(size=1,alpha=0.5) +
        theme_classic(base_size=25) + scale_color_manual(values=CITE_colors) +
        theme(legend.position="none")
print(g)

if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_snATAC_state_UMAP.png'),
                            plot=g,units='in',height=6,width=6,dpi=600)
```

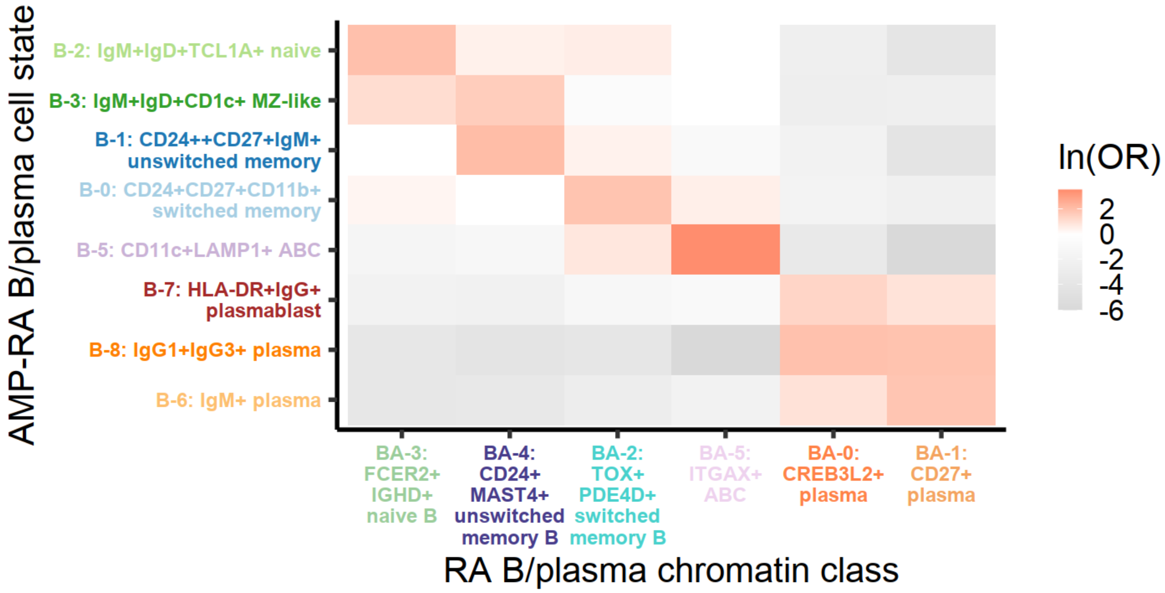


In [17]:

```
#setting order
class_conv_df <- unique(ATAC_meta[,c('cluster_name','cluster_abbr')])
rownames(class_conv_df) <- class_conv_df$cluster_abbr
full_class_order <- class_conv_df[class_order,'cluster_name']

class_state_df$class <- factor(class_state_df$class,levels=class_order)
state_order <- class_state_df[order(class_state_df$class,class_state_df$intOrd),'state']

state_conv_df <- unique(ATAC_meta[,c('CITE','CITE_abbr')])
rownames(state_conv_df) <- state_conv_df$CITE_abbr
full_state_order <- state_conv_df[state_order,'CITE']
```

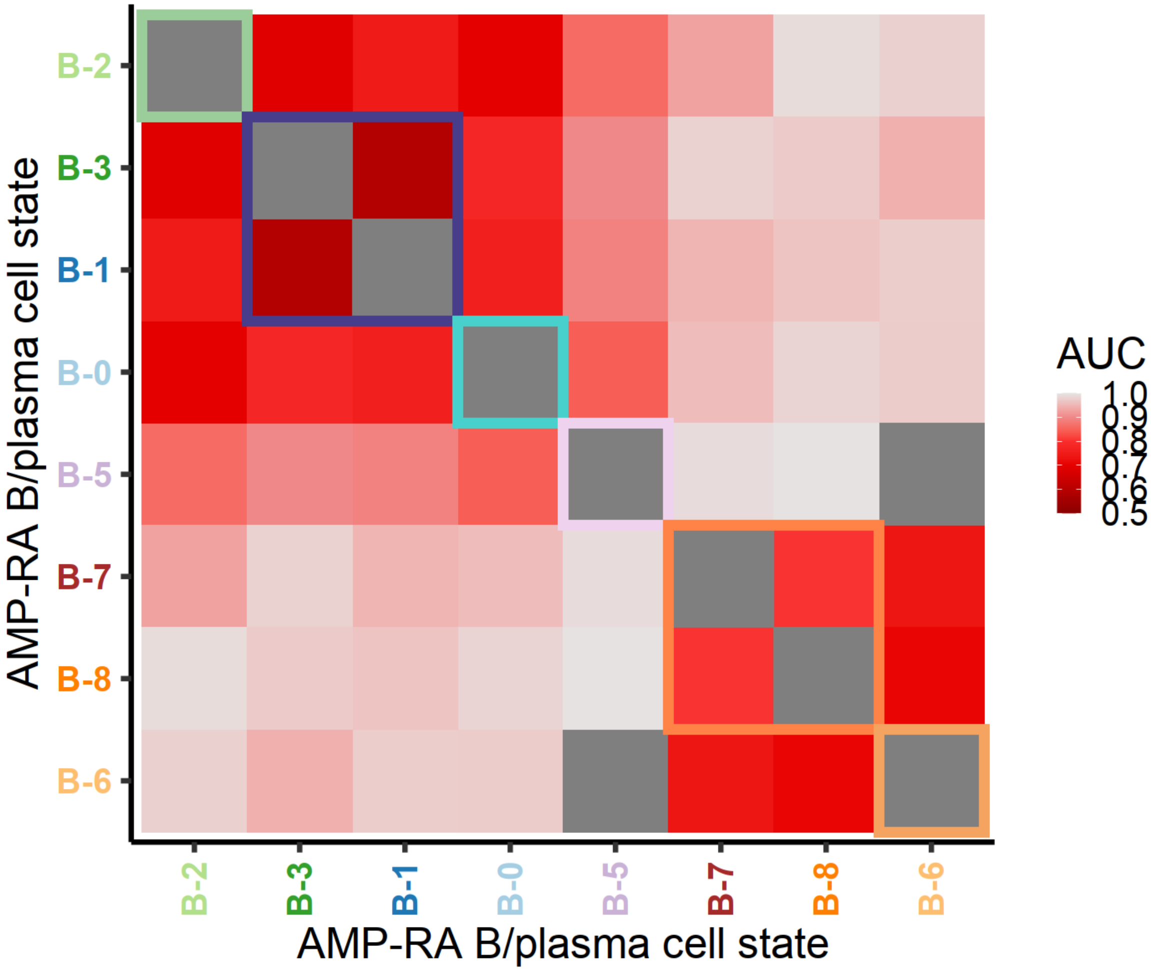
In [18]:

```
#Fig S8g right

#only 9 B-4 cells, so excluding here
fisher_df <- calc_OR(ATAC_meta[which(ATAC_meta$assay=='snATAC' & ATAC_meta$CITE_abbr!='B-4'),], 'cluster_name', 'CITE')
write.table(fisher_df[,c('cluster_name','CITE','OR','pval','padj','CI_low','CI_high')],
            file=paste(sep='',save_dir,CT,'_class_state_OR_table.txt'),quote=FALSE,sep='\t',row.names=FALSE)


g <- plot_OR(fisher_df, 'cluster_name', 'CITE',
             paste('RA',CT_label,'chromatin class'), paste('AMP-RA',CT_label,'cell state'),
             full_class_order, full_state_order,
             clustColors=c(ATAC_colors,CITE_colors),yLab_charLim=27)

options(repr.plot.height=6,repr.plot.width=12)
print(g)

if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_class_state_OR_heatmap.png'),
                            plot=g,units='in',height=6,width=12,dpi=600)
```



In [19]:

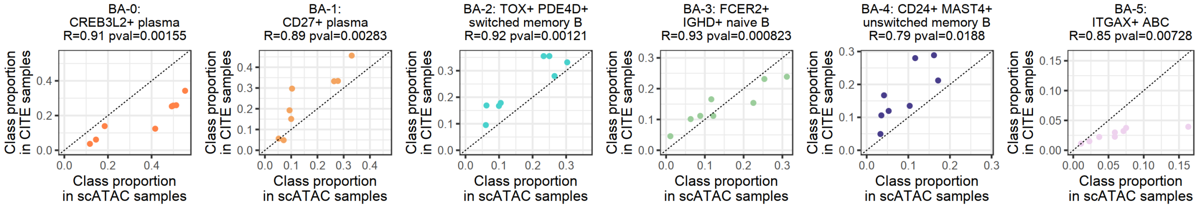
```
#Fig S11d

options(repr.plot.height=10,repr.plot.width=12)
g <- LDA_plots(LDA_res,CT,paste('AMP-RA',CT_label,'cell state'),
               class_state_df=class_state_df,ctOrd_col='intOrd',ctOrd=class_order,
               clustColors=c(CITE_colors,ATAC_colors))
print(g)

if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_LDA_heatmap.png'),
                            plot=g,units='in',height=10,width=12,dpi=600)
```



CITE donor proportions

In [20]:

```
#Fig S12a

tVec <- lapply(sort(unique(ATAC_meta$cluster_name)),replace_space_newline_afterHalf,wiggle=5)
names(tVec) <- sort(unique(ATAC_meta$cluster_name))

options(repr.plot.height=4.25,repr.plot.width=4*length(unique(ATAC_meta$cluster_abbr)))
g <- donor_prop_comp_plot(ATAC_CITE_conv_df,ATAC_meta[which(ATAC_meta$assay=='scATAC'),],CITE_meta,
                          clustColors=ATAC_colors,tSize=18,tVec=tVec)
grid.draw(g)

if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_ATAC_CITE_donor_prop.png'),
                            plot=g,units='in',height=4.25,width=4*length(unique(ATAC_meta$cluster_abbr)),dpi=600)
```



Session Info

```
In [21]: sessionInfo()
```

```
R version 3.6.1 (2019-07-05)
Platform: x86_64-conda_cos6-linux-gnu (64-bit)
Running under: Red Hat Enterprise Linux Server release 6.5 (Santiago)

Matrix products: default
BLAS/LAPACK: /PHShome/kew47/miniconda3/lib/R/lib/libRblas.so

locale:
[1] en_US.UTF-8

attached base packages:
[1] grid      stats     graphics  grDevices utils     datasets  methods  
[8] base     

other attached packages:
 [1] repr_1.0.1        gridExtra_2.3     scales_1.1.1      viridis_0.5.
1
 [5] viridisLite_0.3.0 ggrepel_0.8.2     ggrastr_0.2.3     ggplot2_3.3.
0
 [9] tidyr_1.0.3       stringr_1.4.0     ROCR_1.0-7        gplots_3.0.
1.1
[13] Rmisc_1.5.1       plyr_1.8.6        lattice_0.20-41   gtools_3.8.2
[17] Matrix_1.2-18    

loaded via a namespace (and not attached):
 [1] pbdZMQ_0.3-3        beeswarm_0.2.3      tidyselect_1.1.0   
 [4] purrr_0.3.4         colorspace_1.4-1    vctrs_0.3.5        
 [7] generics_0.0.2      htmltools_0.4.0     base64enc_0.1-3    
[10] rlang_0.4.8         hexbin_1.28.1       pillar_1.4.4       
[13] glue_1.4.0          withr_2.2.0         uuid_0.1-2         
[16] lifecycle_0.2.0     munsell_0.5.0       gtable_0.3.0       
[19] caTools_1.18.0      evaluate_0.14       labeling_0.3       
[22] Cairo_1.5-10        vipor_0.4.5         IRdisplay_0.7.0    
[25] Rcpp_1.0.4.6        KernSmooth_2.23-15  gdata_2.18.0       
[28] IRkernel_1.0.2.9000 jsonlite_1.7.1      farver_2.0.3       
[31] digest_0.6.25       stringi_1.4.6       dplyr_1.0.2        
[34] tools_3.6.1         bitops_1.0-6        magrittr_1.5       
[37] tibble_3.0.1        crayon_1.3.4        pkgconfig_2.0.3    
[40] ellipsis_0.3.1      ggbeeswarm_0.6.0    R6_2.4.1           
[43] compiler_3.6.1     
```

```
In [ ]:
```