Stromal figures

inputs

In [1]:
```
source('jupyterFunctions_perCellType.R')
```

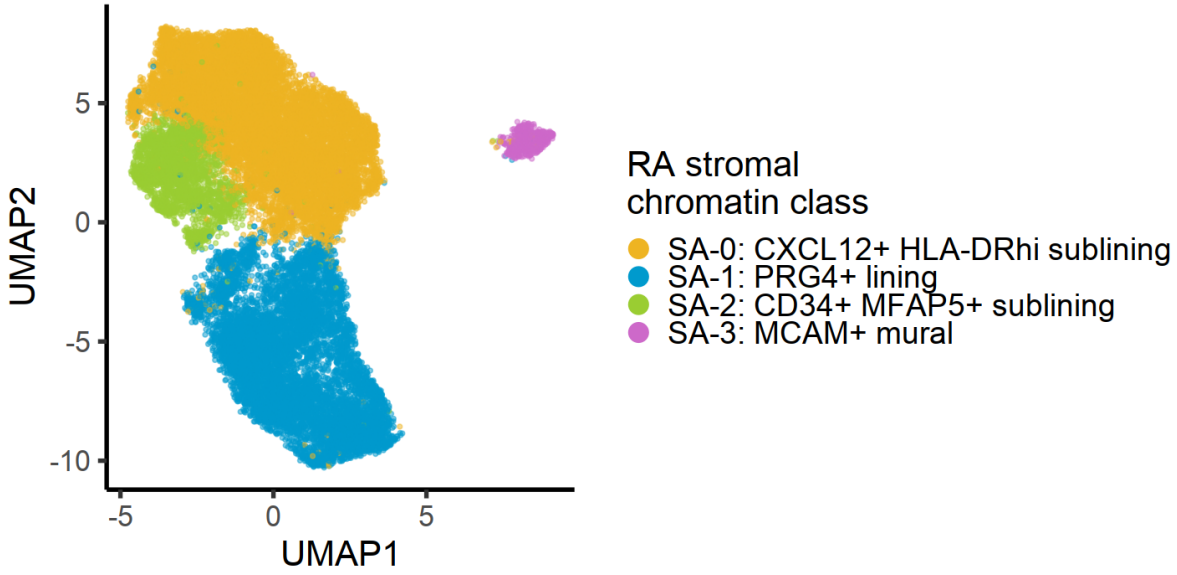
In [2]:
```
CT <- 'stromal'
CT_label <- 'stromal'
data_prefix <- paste(sep='','../data/',CT,'/',CT)
ATAC_meta <- readRDS(paste(sep='',data_prefix,'_ATAC_meta.rds'))
chosenPeaks <- readRDS(paste(sep='',data_prefix,'_chosenPeaks.rds'))
snATAC_pxc_norm <- readRDS(paste(sep='',data_prefix,'_snATAC_pxc_norm.rds'))
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'))
other_resol <- readRDS(paste(sep='',data_prefix,'_ATAC_otherRes.rds'))
CNA_CF <- readRDS(paste(sep='',data_prefix,'_CNA_CTAP-F.rds'))

DNAmethyl_vec <- readRDS(paste(sep='',data_prefix,'_DNAmethylation.rds'))
ATAC_pxc_norm <- readRDS(paste(sep='',data_prefix,'_ATAC_pxc_norm.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')
```

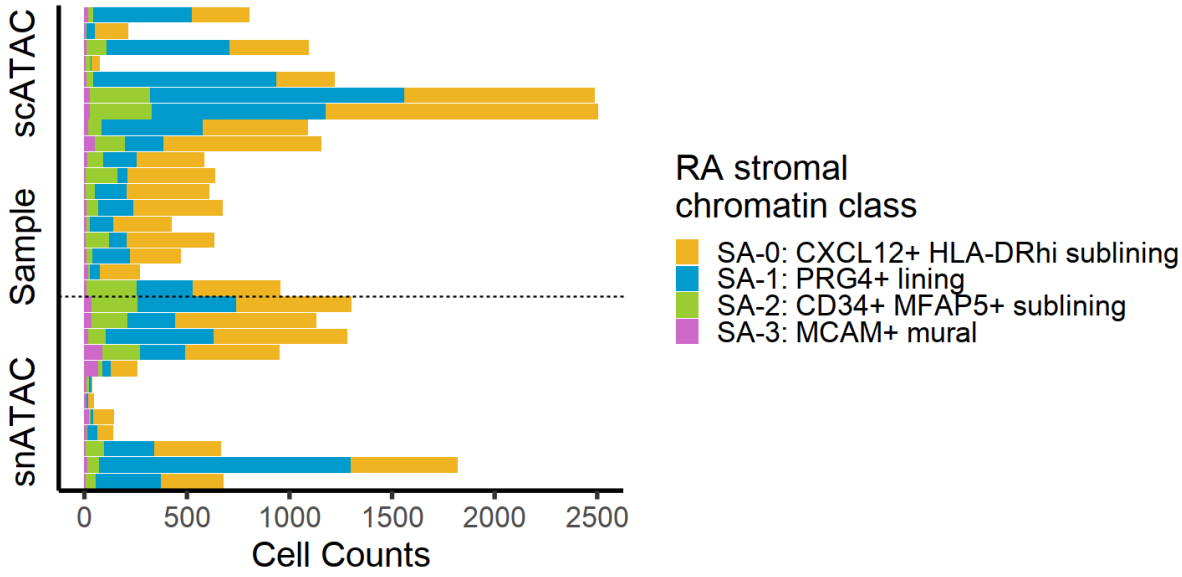
In [27]:
```
save_dir <- NA #'../output/' #or NA if don't want to save
```

ATAC classes

In [5]:
```
#Fig 3a
options(repr.plot.height=6,repr.plot.width=12)
g <- ggplot(ATAC_meta,aes_string(x='UMAP1',y='UMAP2',color='cluster_nam
e')) + geom_point(size=1,alpha=0.5) +
        theme_classic(base_size=25) + scale_color_manual(values=ATAC_col
ors) +
        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_U
MAP.png'),
                            plot=g,units='in',height=6,width=12,dpi=600)
```



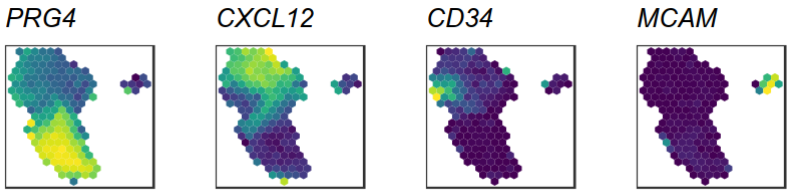
In [6]:
```
#Fig S3a
options(repr.plot.height=6,repr.plot.width=12)
g <- cellCount_bySample_barPlot(ATAC_meta,'sample','cluster_name',paste
(sep='','RA ',CT_label,'\nchromatin class'),
                                ATAC_colors)
print(g)

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



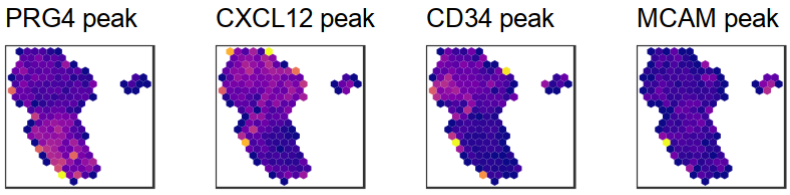
ATAC cluster markers

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

In [8]:
```
#Fig 3b bottom

genes_forUMAPs <- c('PRG4','CXCL12','CD34','MCAM')
if(!all(genes_forUMAPs %in% names(chosenPeaks))) stop('Genes for UMAP no
t in chosen genes')

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

options(repr.plot.height=2,repr.plot.width=7)
g <- plot_markerPeaks_norm_hex_v2(ATAC_meta[multiome_cells,],snRNA_gxc_n
orm[genes_forUMAPs,multiome_cells],'UMAP1','UMAP2',
                                  plot_genes=genes_forUMAPs,plotCol=leng
th(genes_forUMAPs),
                                  titleSize=15,hex_bins=17,cutCap=0)
grid.draw(g)
if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_markerGene_U
MAP.png'),
                            plot=g,units='in',height=2,width=7,dpi=600)
```



In [9]:
```
#Fig 3b top

toPlot <- snATAC_pxc_norm[unname(chosenPeaks[genes_forUMAPs]),multiome_c
ells]
rownames(toPlot) <- paste(sep='',names(chosenPeaks[genes_forUMAPs]),' pe
ak')

options(repr.plot.height=2,repr.plot.width=7)
g <- plot_markerPeaks_norm_hex_v2(ATAC_meta[multiome_cells,],toPlot,'UMA
P1','UMAP2',
                                  plot_genes=rownames(toPlot),plotCol=nr
ow(toPlot),titleSize=15,hex_bins=17,cutCap=0,
                                  titleFace='plain',colorOpt='plasma')
grid.draw(g)
if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_markerPeak_U
MAP.png'),
                            plot=g,units='in',height=2,width=7,dpi=600)
```



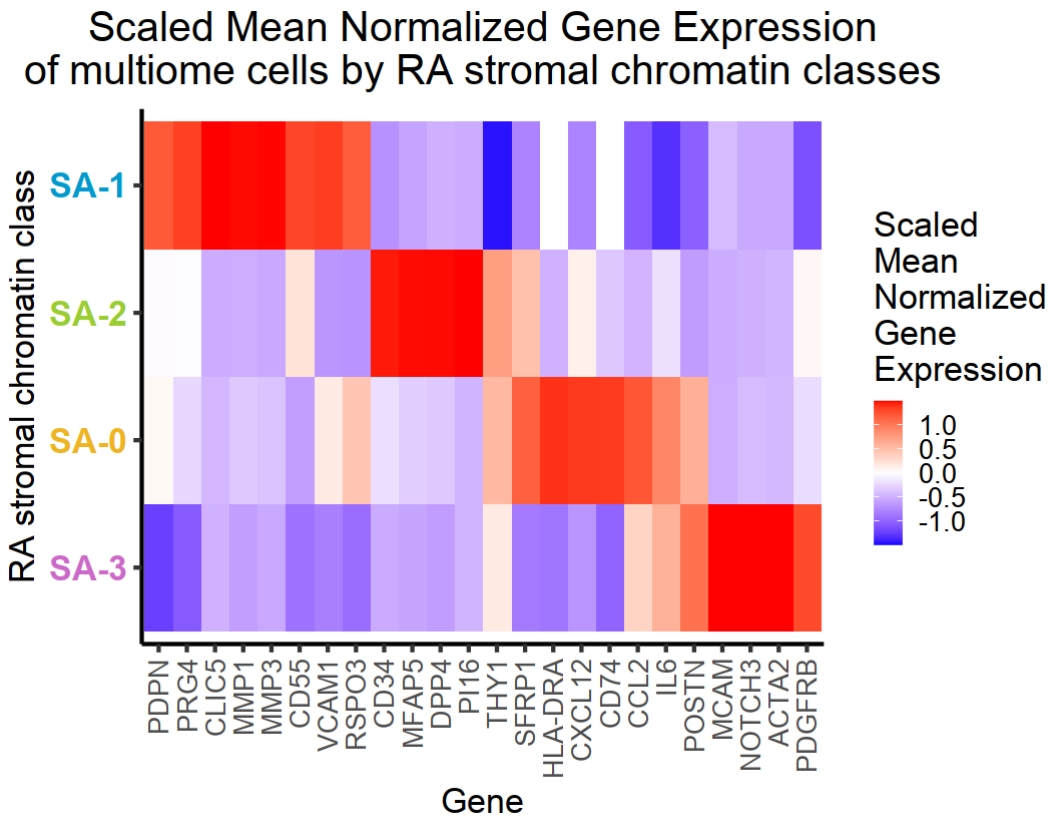
In [10]:
```
class_order <- c('SA-1','SA-2','SA-0','SA-3')
all(class_order %in% ATAC_meta$cluster_abbr)
```

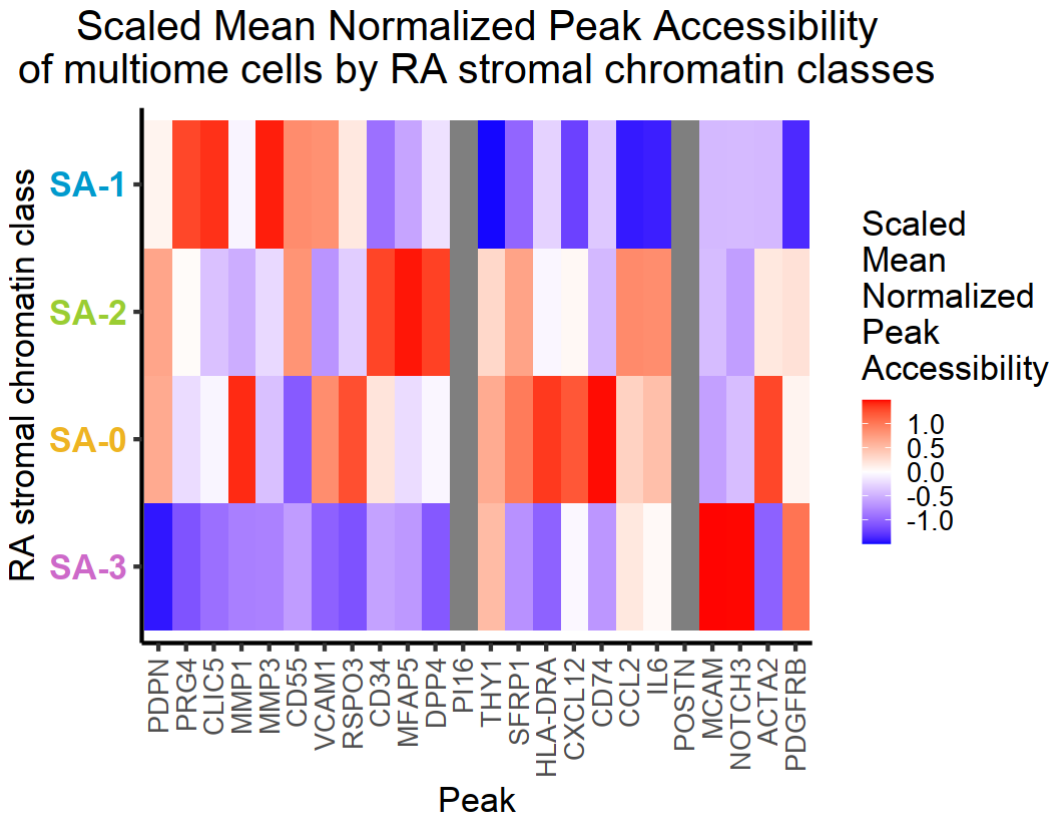
TRUE

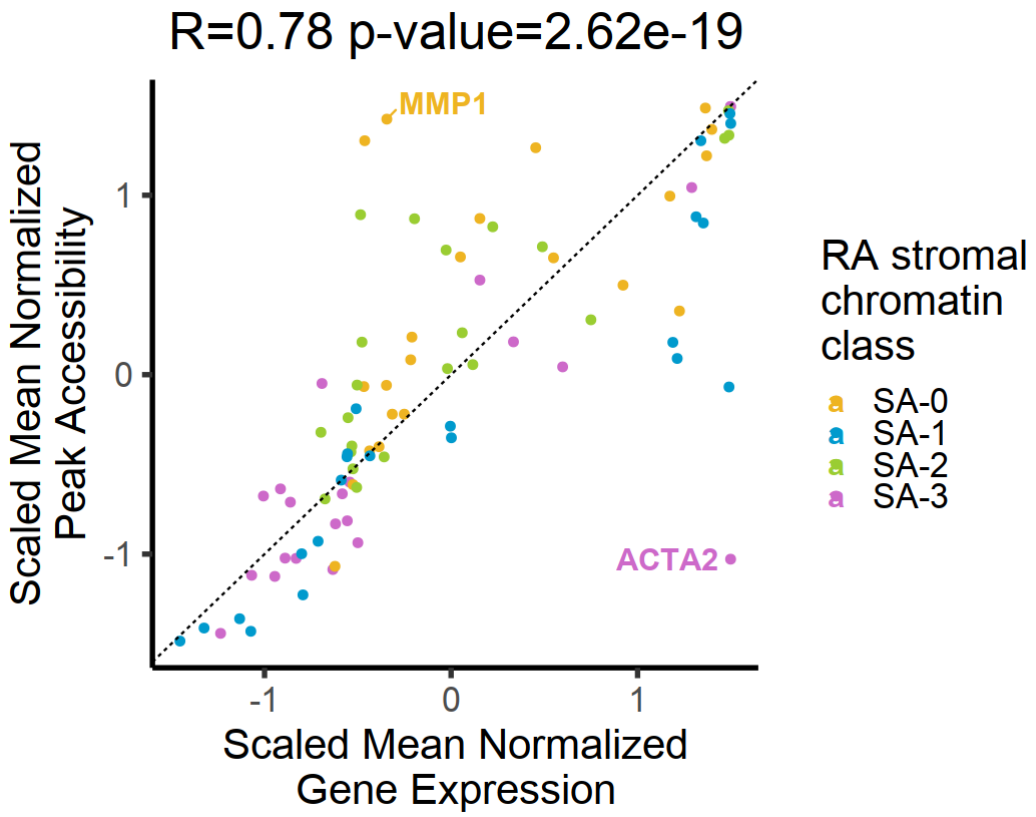
In [11]:

```
#Fig S3b

res <- scaleFeat_forHeatmap(chosenGenes,class_order,chosenPeaks,snRNA_gx
CT_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',past
e('RA',CT_label,'chromatin class'),
                               'Scaled\nMean\nNormalized\nGene\nExpressi
on',
                               plotTit=paste('Scaled Mean Normalized Gen
e Expression\nof multiome cells by RA',
                                             CT_label,'chromatin classe
s'),
                               scale_lim=scale_lim,clustColors=ATAC_colo
rs)
print(g)
if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_markerGene_h
eatmap.png'),
                            plot=g,units='in',height=7,width=9,dpi=600)

g <- pseudobulk_scaled_heatmap(snATAC_pxCT_norm_subset_scaled,'Peak',pas
te('RA',CT_label,'chromatin class'),
                               'Scaled\nMean\nNormalized\nPeak\nAccessib
ility',
                               plotTit=paste('Scaled Mean Normalized Pea
k Accessibility\nof multiome cells by RA',
                                             CT_label,'chromatin classe
s'),
                               scale_lim=scale_lim,clustColors=ATAC_colo
rs)
print(g)
if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_markerPeak_h
eatmap.png'),
                            plot=g,units='in',height=7,width=9,dpi=600)

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

fxCT_norm_subset_scaled$label <- ''
fxCT_norm_subset_scaled[which(fxCT_norm_subset_scaled$gene=='MMP1' & fxC
T_norm_subset_scaled$cluster_abbr=='SA-0'),
                        'label'] <- 'MMP1'
fxCT_norm_subset_scaled[which(fxCT_norm_subset_scaled$gene=='ACTA2' & fx
CT_norm_subset_scaled$cluster_abbr=='SA-3'),
                        'label'] <- 'ACTA2'

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 stromal chromatin classes
RA stromal chromatin class
SA-1
SA-2
SA-0
SA-3
PDPN
PRG4
CLIC5
MMP1
MMP3
CD55
VCAM1
RSPO3
CD34
MFAP5
DPP4
PI16
THY1
SFRP1
HLA-DRA
CXCL12
CD74
CCL2
IL6
POSTN
MCAM
NOTCH3
ACTA2
PDGFRB
Gene
Scaled Mean Normalized Gene Expression
1.0
0.5
0.0
-0.5
-1.0

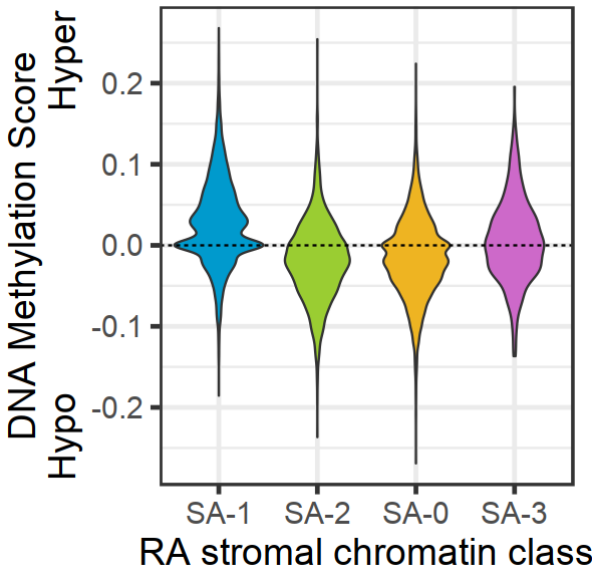

Scaled Mean Normalized Peak Accessibility
of multiome cells by RA stromal chromatin classes
RA stromal chromatin class
SA-1
SA-2
SA-0
SA-3
PDPN
PRG4
CLIC5
MMP1
MMP3
CD55
VCAM1
RSPO3
CD34
MFAP5
DPP4
PI16
THY1
SFRP1
HLA-DRA
CXCL12
CD74
CCL2
IL6
POSTN
MCAM
NOTCH3
ACTA2
PDGFRB
Peak
Scaled Mean Normalized Peak Accessibility
1.0
0.5
0.0
-0.5
-1.0


R=0.78 p-value=2.62e-19
MMP1
ACTA2
Scaled Mean Normalized Peak Accessibility
Scaled Mean Normalized Gene Expression
1
0
-1
-1
0
1
RA stromal chromatin class
SA-0
SA-1
SA-2
SA-3

DNA Methylation Score

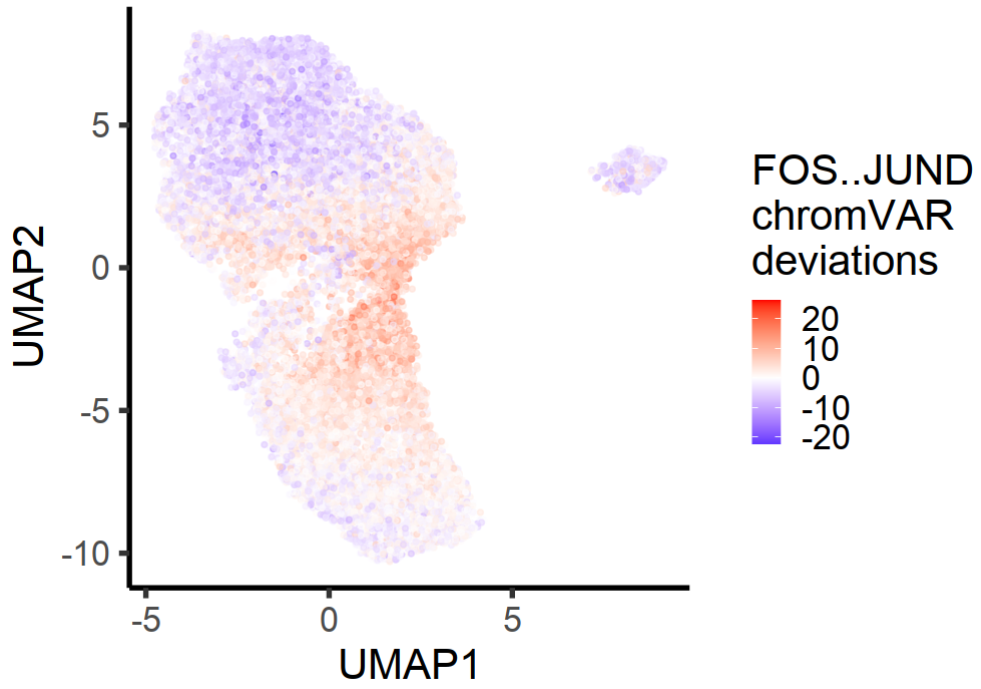
In [12]:
```
#Fig S3c

hyper_assoc_peaks <- names(DNAmethyl_vec[which(DNAmethyl_vec=='hypermeth
ylated')])
hypo_assoc_peaks <- names(DNAmethyl_vec[which(DNAmethyl_vec=='hypomethyl
ated')])

perCell_scores <- ATAC_perCell_score(ATAC_pxc_norm,hyper_assoc_peaks,hyp
o_assoc_peaks)

toPlot <- cbind(ATAC_meta, 'DNAmethyl_score'=perCell_scores[rownames(ATA
C_meta)])
toPlot$cluster_abbr <- factor(toPlot$cluster_abbr,levels=class_order)

options(repr.plot.height=5,repr.plot.width=5)
g <- ggplot(toPlot,aes_string(x='cluster_abbr',y='DNAmethyl_score',fill
='cluster_name')) + geom_violin() +
        theme_bw(base_size=22) + scale_fill_manual(values=ATAC_colors) +
theme(legend.position="none") +
        labs(x=paste('RA',CT_label,'chromatin class'),
             y='DNA Methylation Score\nHypo                                H
yper') +
        geom_hline(yintercept=0,linetype='dashed',color='black')
print(g)

if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_DNAmethylati
on_scores.png'),
                            plot=g,units='in',height=5,width=5,dpi=600)


within <- toPlot[which(toPlot$cluster_abbr=='SA-0'),'DNAmethyl_score']
without <- toPlot[which(toPlot$cluster_abbr!='SA-0'),'DNAmethyl_score']

ll <- wilcox.test(within,without,alternative = "less")
ll
ll$p.value
```

```
        Wilcoxon rank sum test with continuity correction

data:  within and without
W = 48460654, p-value < 2.2e-16
alternative hypothesis: true location shift is less than 0

0
```



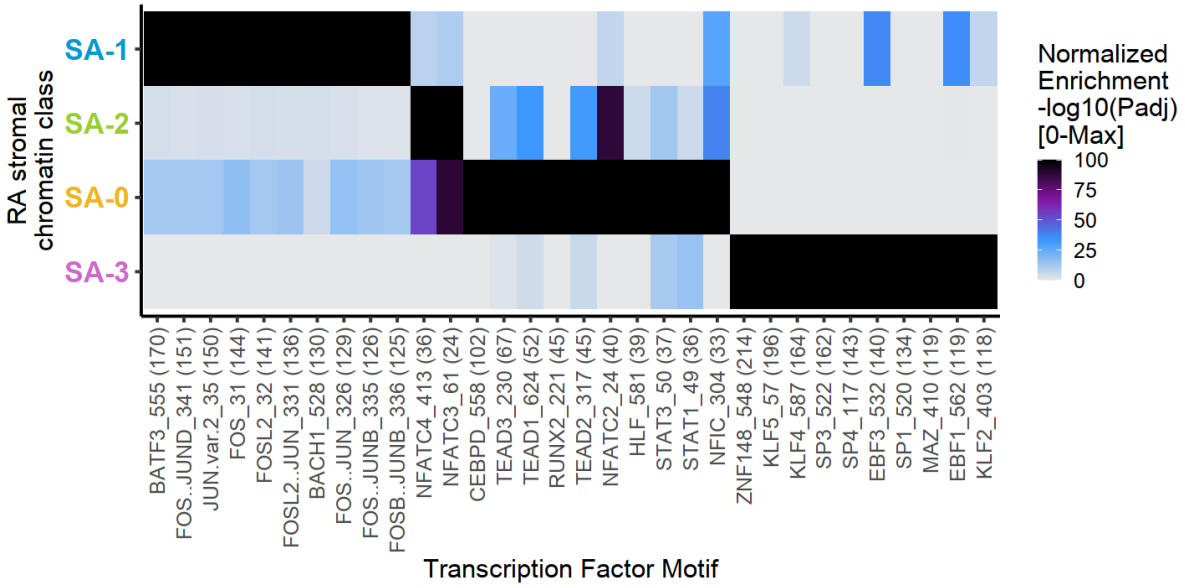
TFs

In [13]:
```
#Fig 3c 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])
if(!identical(sort(new_colnames),sort(rownames(ATAC_meta)))) stop('celln
ames not consistent b/t ATAC_meta and chromVAR')
colnames(chromVARz_mat) <- new_colnames

motif_toPlot <- 'FOS..JUND_341'
toPlot <- cbind(ATAC_meta,'motif'=chromVARz_mat[motif_toPlot,rownames(AT
AC_meta)])

options(repr.plot.height=6,repr.plot.width=8.5)
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\ndeviations'))
print(g)

if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_motif_',str_
replace(motif_toPlot,'\\.\\.','-'),'_UMAP.png'),
                            plot=g,units='in',height=6,width=8.5,dpi=60
0)
```



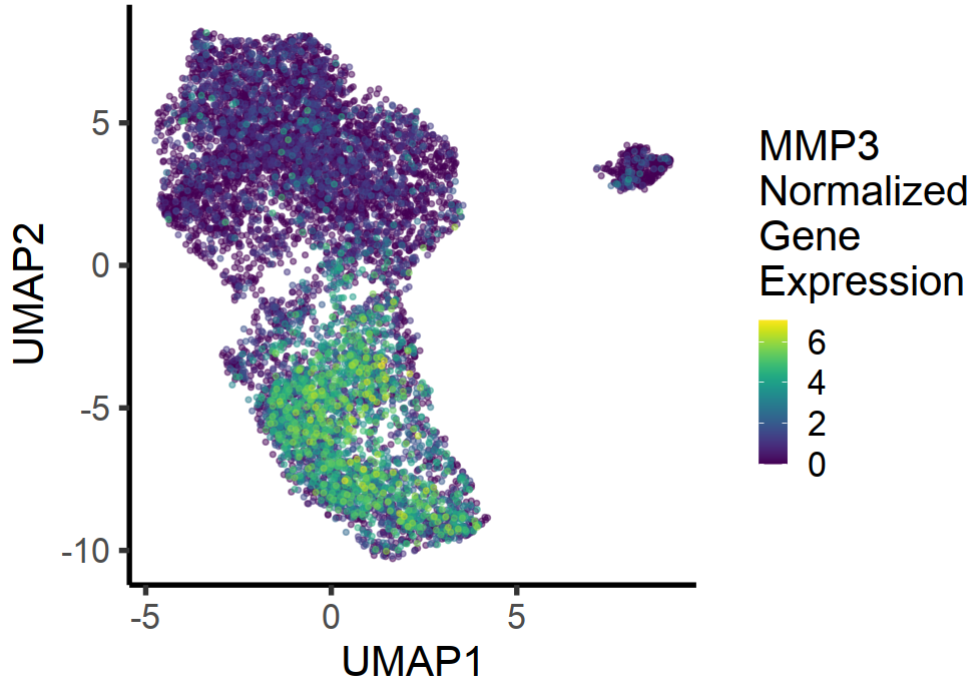
In [14]:

```
#Fig 3c 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=10,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)
```



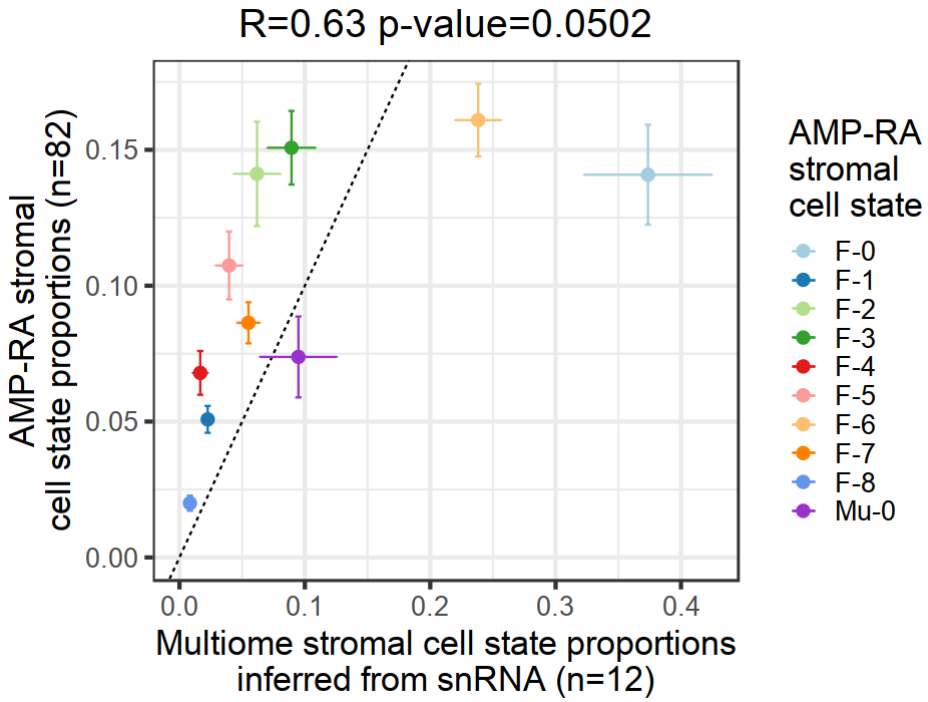
In [15]:

```
#Fig 3d left

gene_toPlot <- 'MMP3'
toPlot <- cbind(ATAC_meta[multiome_cells,],'gene'=snRNA_gxc_norm[gene_to
Plot,multiome_cells])

options(repr.plot.height=6,repr.plot.width=8.5)
g <- ggplot(toPlot[order(toPlot$gene),],aes_string(x='UMAP1',y='UMAP2',c
olor='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=8.5,dpi=60
0)
```

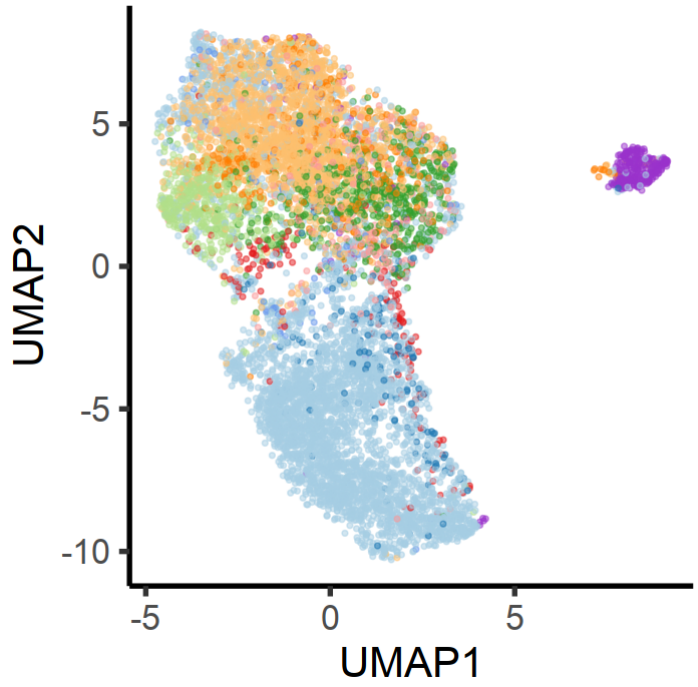


Transcriptional Cell States

In [16]:
```
#Fig S8b

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)
```



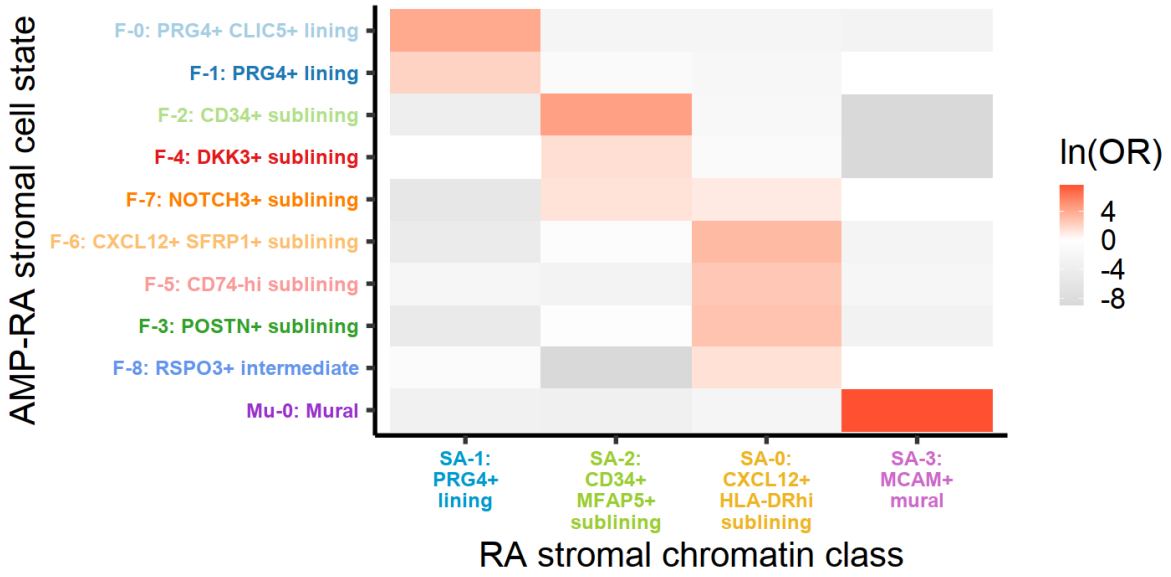
In [17]:
```
#Fig 7b left
options(repr.plot.height=6,repr.plot.width=6)
g <- ggplot(ATAC_meta[multiome_cells,],aes_string(x='UMAP1',y='UMAP2',co
lor='CITE')) +
        geom_point(size=1,alpha=0.5) +
        theme_classic(base_size=25) + scale_color_manual(values=CITE_col
ors) +
        theme(legend.position="none")
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 [18]:
```
#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']
```

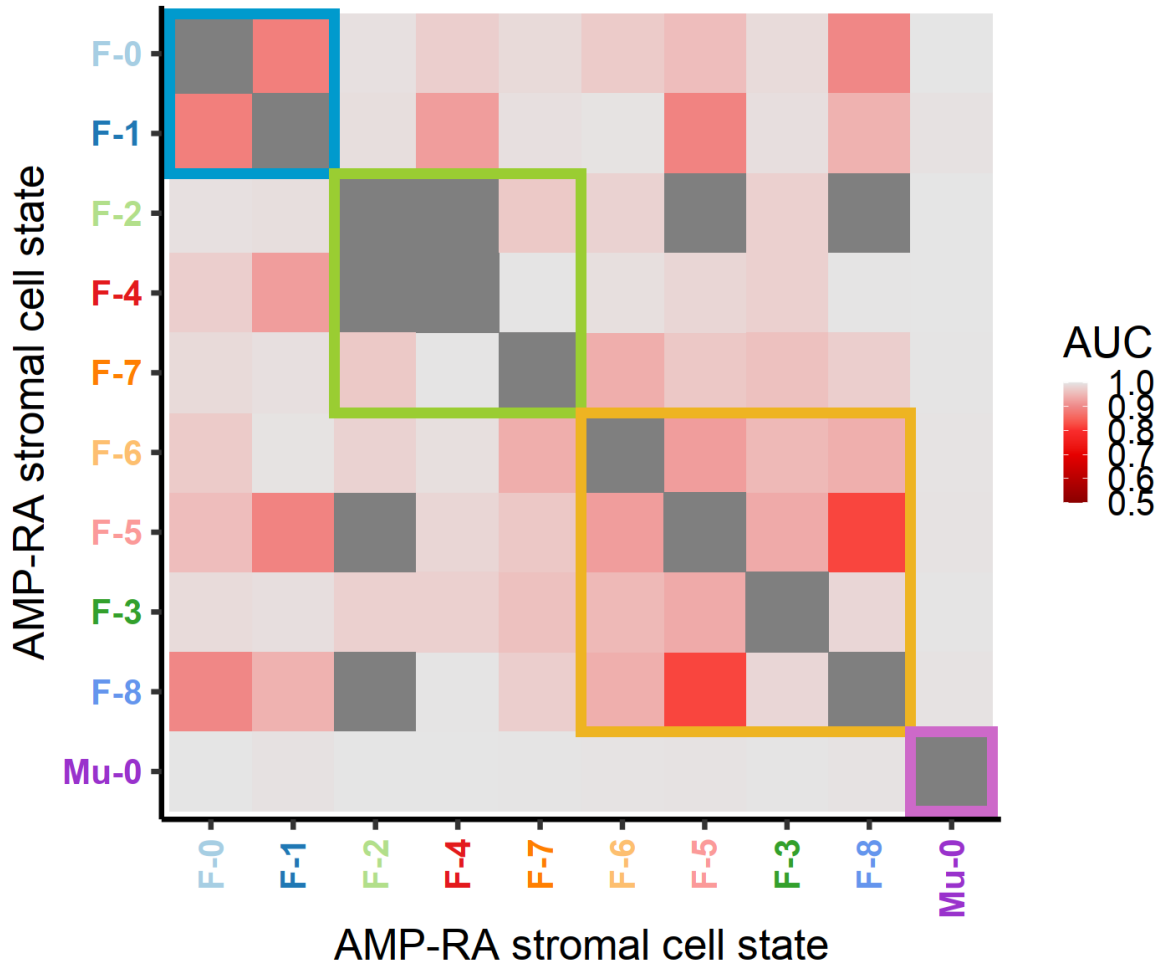
In [19]:

```
#Fig 7b right

fisher_df <- calc_OR(ATAC_meta[multiome_cells,], '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))

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)
```

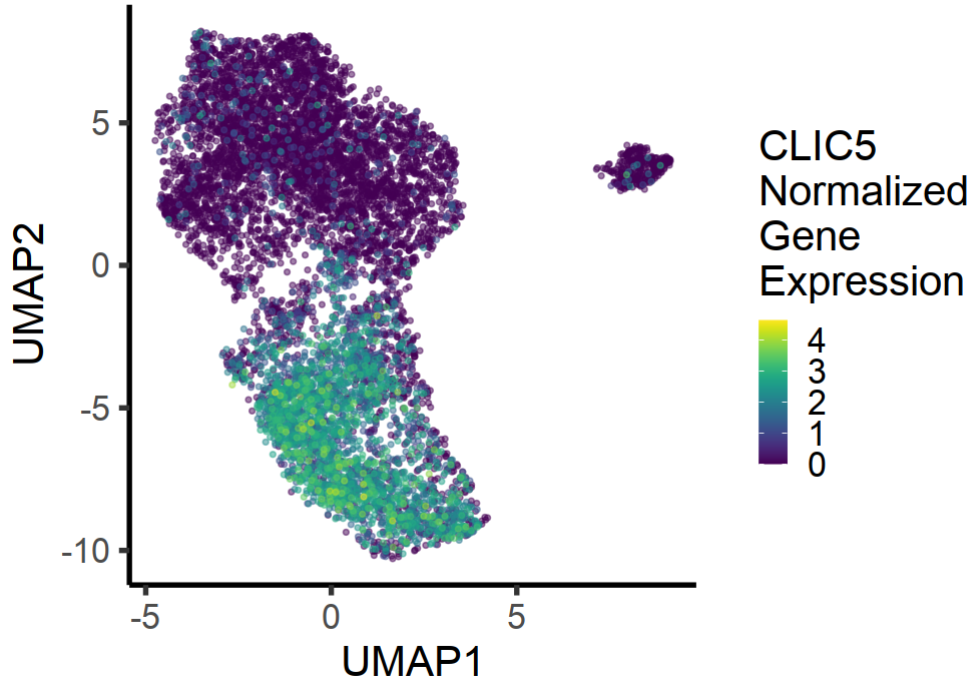


In [20]:

```
#Fig S11b

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=cl
ass_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=60
0)
```

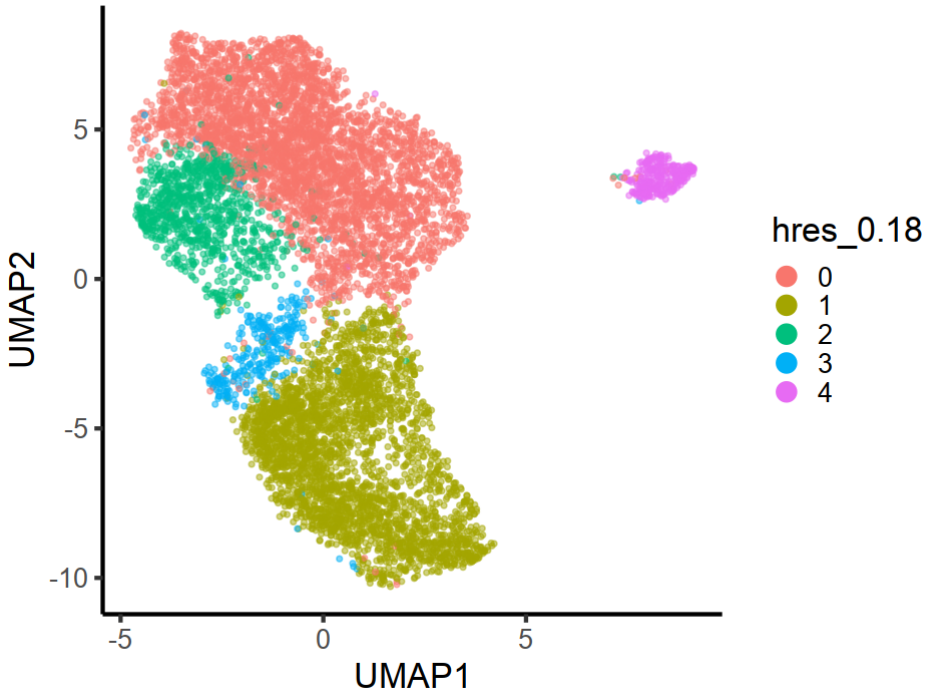


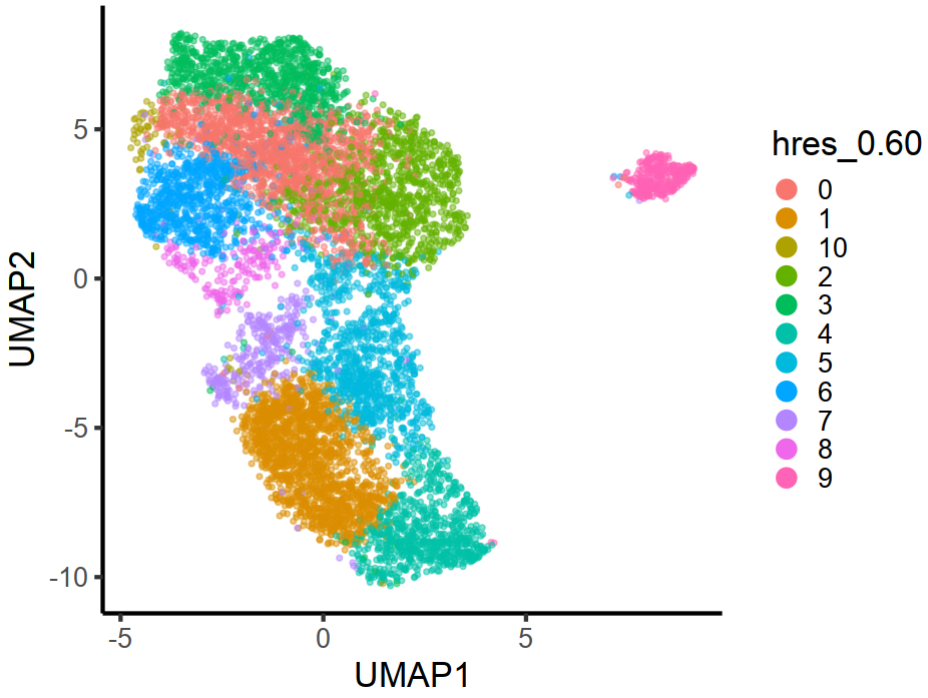
In [21]:
```
#Fig S9b left

gene_toPlot <- 'CLIC5'
toPlot <- cbind(ATAC_meta[multiome_cells,],'gene'=snRNA_gxc_norm[gene_to
Plot,multiome_cells])

options(repr.plot.height=6,repr.plot.width=8.5)
g <- ggplot(toPlot[order(toPlot$gene),],aes_string(x='UMAP1',y='UMAP2',c
olor='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=8.5,dpi=60
0)
```



additional resolutions

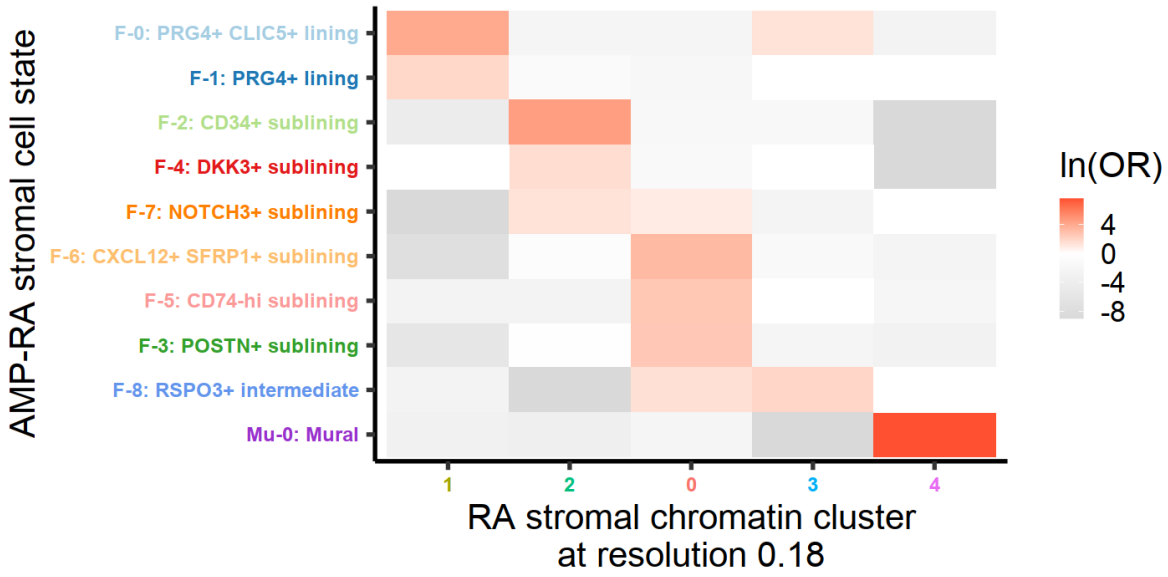
In [22]:
```
#Fig S10b top
if(!identical(sort(rownames(other_resol)),sort(rownames(ATAC_meta)))) st
op('rownames need to be the same')
toPlot <- cbind(ATAC_meta,other_resol[rownames(ATAC_meta),])

for(cc in colnames(other_resol)){

    cluster_colors <- hue_pal()(length(unique(toPlot[,cc])))
    names(cluster_colors) <- sort(unique(toPlot[,cc]))

    options(repr.plot.height=6,repr.plot.width=8)
    g <- ggplot(toPlot[multiome_cells,],aes_string(x='UMAP1',y='UMAP2',c
olor=cc)) + geom_point(size=1,alpha=0.5) +
            theme_classic(base_size=20) + scale_color_manual(values=clus
ter_colors) +
            guides(colour = guide_legend(override.aes = list(size=5,alph
a=1)))
    print(g)

    if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_ATAC_',c
c,'_UMAP.png'),
                                plot=g,units='in',height=6,width=8,dpi=6
00)

}
```

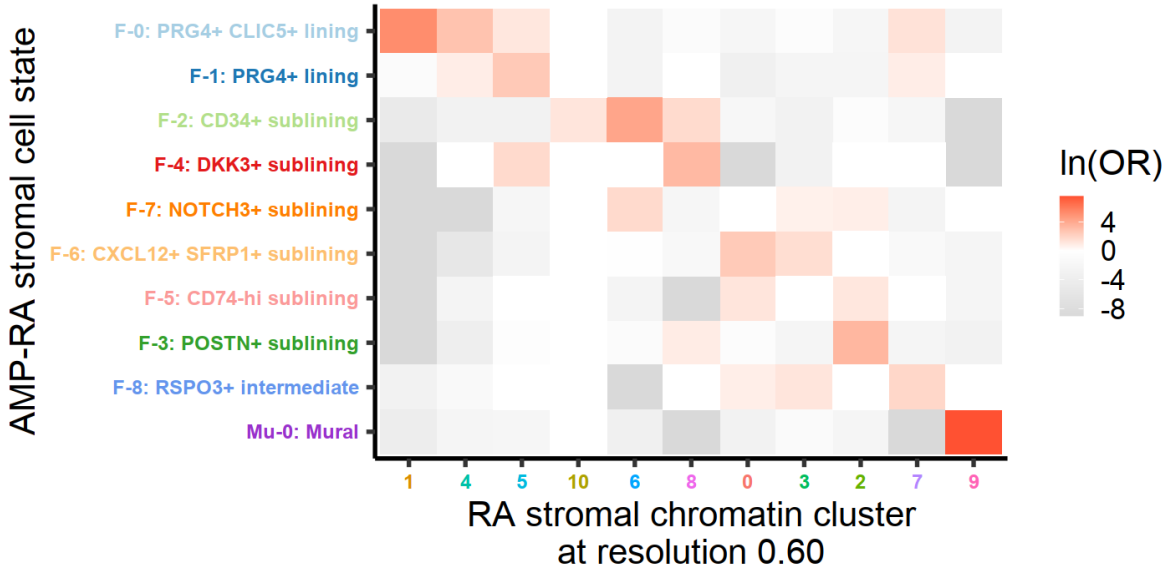

UMAP2
5
0
-5
-10
-5
0
5
UMAP1
hres_0.18
0
1
2
3
4


UMAP2
5
0
-5
-10
-5
0
5
UMAP1
hres_0.60
0
1
10
2
3
4
5
6
7
8
9

In [23]:

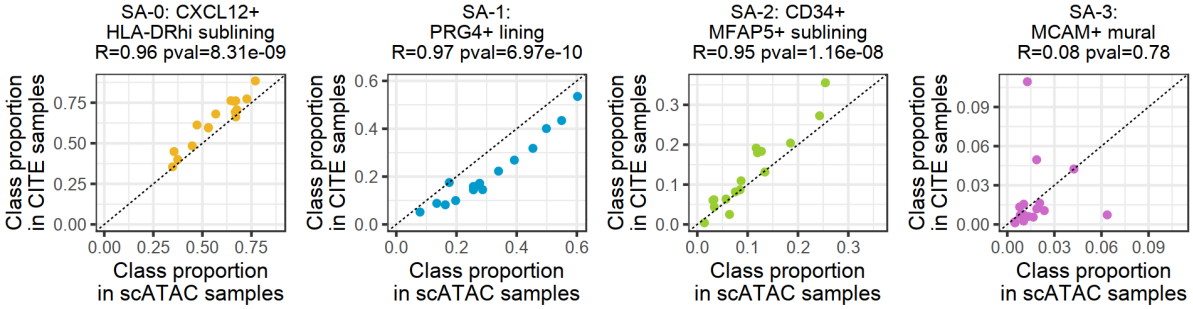
```
#Fig S10b bottom
if(!identical(sort(rownames(other_resol)),sort(rownames(ATAC_meta)))) st
op('rownames need to be the same')
toPlot <- cbind(ATAC_meta,other_resol[rownames(ATAC_meta),])

for(cc in colnames(other_resol)){

    cluster_colors <- hue_pal()(length(unique(toPlot[,cc])))
    names(cluster_colors) <- sort(unique(toPlot[,cc]))

    fisher_df <- calc_OR(toPlot[multiome_cells,], cc, 'CITE')
    this_state_order <- full_state_order
    clustOrd <- reorder_col_diag_plotOR(fisher_df,cc,'CITE',yOrd=this_st
ate_order,mCol='lnOR',op='max')

    g <- plot_OR(fisher_df, cc, 'CITE',
                 paste('RA',CT_label,'chromatin cluster\nat resolution',
str_split_fixed(cc,'_',2)[,2]),
                 paste('AMP-RA',CT_label,'cell state'),
                 clustOrd, this_state_order,
                 clustColors=c(cluster_colors,CITE_colors))

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

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


AMP-RA stromal cell state
F-0: PRG4+ CLIC5+ lining
F-1: PRG4+ lining
F-2: CD34+ sublining
F-4: DKK3+ sublining
F-7: NOTCH3+ sublining
F-6: CXCL12+ SFRP1+ sublining
F-5: CD74-hi sublining
F-3: POSTN+ sublining
F-8: RSPO3+ intermediate
Mu-0: Mural
1
2
0
3
4
RA stromal chromatin cluster
at resolution 0.18
ln(OR)
4
0
-4
-8


AMP-RA stromal cell state
F-0: PRG4+ CLIC5+ lining
F-1: PRG4+ lining
F-2: CD34+ sublining
F-4: DKK3+ sublining
F-7: NOTCH3+ sublining
F-6: CXCL12+ SFRP1+ sublining
F-5: CD74-hi sublining
F-3: POSTN+ sublining
F-8: RSPO3+ intermediate
Mu-0: Mural
1
4
5
10
6
8
0
3
2
7
9
RA stromal chromatin cluster
at resolution 0.60
ln(OR)
4
0
-4
-8

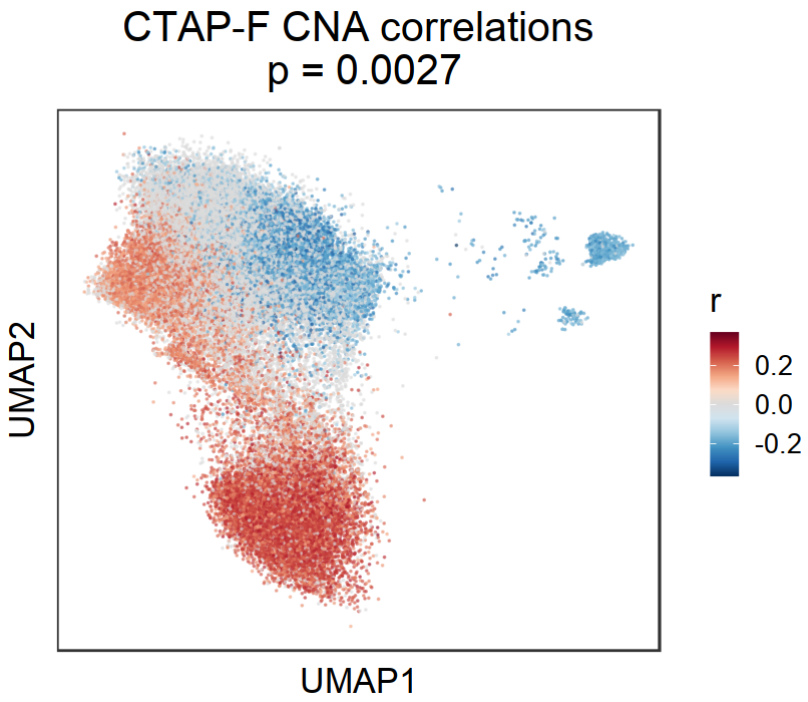
CITE donor proportions

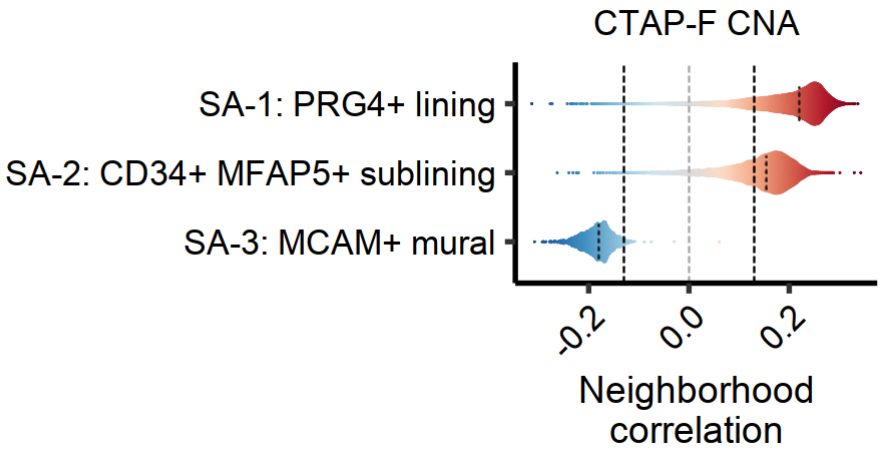
In [24]:
```
#Fig S12a

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

options(repr.plot.height=4.25,repr.plot.width=4*length(unique(ATAC_meta
$cluster_abbr)))
g <- donor_prop_comp_plot(ATAC_CITE_conv_df,ATAC_meta[which(ATAC_meta$as
say=='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_do
nor_prop.png'),
                            plot=g,units='in',height=4.25,width=4*length
(unique(ATAC_meta$cluster_abbr)),dpi=600)
```



CNA associations

In [25]:
```
#Fig S12c

CNA_CF_FDR <- 0.13
CNA_CF_globalp <- 0.0027
CNA_title <- 'CTAP-F'
CNA_col <- 'cna_corr_F'
class_col <- 'ATAC_cluster_name'

toPlot <- CNA_add_col(CITE_meta, CNA_CF, CNA_col)

options(repr.plot.height=6,repr.plot.width=7)
g <- CNA_umap_plots(toPlot,'ATAC_UMAP1','ATAC_UMAP2',CNA_col,
                    thisTitle=paste(CNA_title,'CNA correlations\n','p
='',CNA_CF_globalp),
                    smallPt=TRUE,fdr_thresh=CNA_CF_FDR)
print(g)
if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_',str_replac
e(CNA_title,' ','_'),'_CNA_UMAP.png'),
                                plot=g,units='in',height=6,width=7,dpi=6
00)

#only plot those classes whose median is above FDR thresholds
ll <- aggregate(toPlot[,CNA_col] ~ toPlot[,class_col], FUN=median)
colnames(ll) <- c(class_col,CNA_col)
CNA_classes <- ll[which(abs(ll[,CNA_col])>CNA_CF_FDR),class_col]


options(repr.plot.height=4,repr.plot.width=8)
g <- CNA_violin_plots(toPlot[which(toPlot[,class_col] %in% CNA_classe
s),],
                      class_col,CNA_col,CNA_CF_FDR,thisTitle=paste(CNA_t
itle,'CNA'))
print(g)
if(!is.na(save_dir)) ggsave(file=paste(sep='',save_dir,CT,'_',str_replac
e(CNA_title,' ','_'),'_CNA_violin.png'),
                                plot=g,units='in',height=4,width=8,dpi=6
00)
```


CTAP-F CNA correlations
p = 0.0027
UMAP2
UMAP1
r
0.2
0.0
-0.2


CTAP-F CNA
SA-1: PRG4+ lining
SA-2: CD34+ MFAP5+ sublining
SA-3: MCAM+ mural
-0.2
0.0
0.2
Neighborhood correlation

Session Info

In [26]:
```
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          RColorBrewer_1.1-2
[16] uuid_0.1-2           lifecycle_0.2.0      munsell_0.5.0
[19] gtable_0.3.0         caTools_1.18.0       evaluate_0.14
[22] labeling_0.3         Cairo_1.5-10         vipor_0.4.5
[25] IRdisplay_0.7.0      Rcpp_1.0.4.6         KernSmooth_2.23-15
[28] gdata_2.18.0         IRkernel_1.0.2.9000  jsonlite_1.7.1
[31] farver_2.0.3         digest_0.6.25        stringi_1.4.6
[34] dplyr_1.0.2          tools_3.6.1          bitops_1.0-6
[37] magrittr_1.5         tibble_3.0.1         crayon_1.3.4
[40] pkgconfig_2.0.3      ellipsis_0.3.1       ggbeeswarm_0.6.0
[43] R6_2.4.1             compiler_3.6.1
```

In [ ]:
```

```