

# snATAC-seq Pilot Data

## Libraries & Functions

```
In [1]: suppressMessages({  
  library(tidyverse)  
  library(stringr)  
  library(rstatix)  
  library(ggplot2)  
  library(ggtext)  
  library(scales)  
  library(ggpubr)  
  library(repr)  
})
```

```
In [2]: plot_fig <- function(toPlot,xCol,yCol,cCol,plotTitle,subTitle,colorTitle,plotColors="NA",order=NA,  
                           NAcolor='grey50',xTitle='ATAC UMAP1',yTitle='ATAC UMAP2',rasterDPI=300,baseSize=25){  
  if(!all(c(xCol,yCol,cCol) %in% colnames(toPlot))) stop('columns not in df')  
  
  if(!is.na(order)){  
    if(order=='desc'){  
      toPlot <- toPlot[order(toPlot[,cCol],decreasing=TRUE),]  
    } else if(order=='asc'){  
      toPlot <- toPlot[order(toPlot[,cCol],decreasing=FALSE),]  
    } else if(order=='rand'){  
      set.seed(0)  
      toPlot <- toPlot[sample(nrow(toPlot),nrow(toPlot)),]  
    } else {  
      cat('order parameter can only be: desc, asc, rand. Not changing cell order.\n')  
    }  
  }  
  
  g <- ggplot(toPlot,aes(x=!!sym(xCol),y=!!sym(yCol),color=!!sym(cCol))) +  
    rasterise(geom_point(size=1,alpha=0.5),dpi=rasterDPI) + theme_classic(base_size=baseSize) +  
    labs(color=colorTitle,title=plotTitle,subtitle=subTitle,x=xTitle,y=yTitle) +  
    theme(plot.title = element_text(hjust = 0.5),plot.subtitle = element_text(hjust = 0.5))
```

```

if(all(unique(toPlot[,cCol]) %in% names(plotColors))){  

  g <- g + scale_color_manual(values=plotColors) +  

    guides(colour = guide_legend(override.aes = list(size=5,alpha=1)))  

} else if(str_count(plotColors,',') == 3){  

  ll <- str_split_fixed(plotColors,',',4)  

  min_lim <- as.numeric(ll[3])  

  max_lim <- as.numeric(ll[4])  

  if(min(toPlot[,cCol],na.rm = TRUE) < min_lim |  

    max(toPlot[,cCol],na.rm = TRUE) > max_lim) cat("WARNING: limits\n")  

  g <- g + scale_color_gradient(low=ll[1],high=ll[2],limits=c(min_lim,max_lim),na.value=NAcolor)  

} else if(str_count(plotColors,',') == 5){  

  ll <- str_split_fixed(plotColors,',',6)  

  min_lim <- as.numeric(ll[5])  

  max_lim <- as.numeric(ll[6])  

  if(min(toPlot[,cCol],na.rm = TRUE) < min_lim |  

    max(toPlot[,cCol],na.rm = TRUE) > max_lim) cat("WARNING: limits\n")  

  mid_pt <- as.numeric(ll[4])  

  if(mid_pt < min_lim | mid_pt > max_lim) stop('ERROR: midpoint')  

  g <- g + scale_color_gradient2(low=ll[1],mid=ll[2],high=ll[3],midpoint=mid_pt,  

    limits=c(min_lim,max_lim),na.value=NAcolor)
}  

return(g)
}

```

## Global variables

```
In [3]: data_dir <- '../data/dataset2/'  

dataset_str <- 'dataset2'  

dataset_str_long <- 'Dataset 2'  

CT_str <- 'RA T Cell State'  

ori_CT_str <- 'RA T Cell Chromatin Class'  

sample_str <- '(n=12 samples)'
```

```
meta <- readRDS(paste(sep='', data_dir, dataset_str, '_metadata.rds'))  
  
colors <- readRDS('../data/misc/RA_colors.rds')
```

```
In [4]: good_color <- 'green'  
midpt_color <- 'snow2'  
cLISI_color <- '#03EEEE'  
iLISI_color <- '#FFCA03'  
sKNN_color <- '#CDAAFE'  
lsKLD_color <- '#FFA7ED'
```

```
In [30]: save_dir <- NA #'../output/' #or NA if don't want to save  
file_extension <- '.pdf'
```

## Basic GAS - PCA

```
In [6]: method <- 'PCA'  
feature <- 'Basic GAS'  
method_feature_prefix <- paste(sep='_', gsub(' ', '', gsub(' \\\+ Harmony', '_Harmony', method)),  
                               gsub(' ', '', feature))  
outFile_prefix <- paste(sep="_", dataset_str, method_feature_prefix)  
  
df <- readRDS(paste(sep='', data_dir, dataset_str, '_pilotData_', method_feature_prefix, '.rds'))  
df$sKNN_log10 <- log10(df$sKNN)  
df[which(is.infinite(df$sKNN_log10)), 'sKNN_log10'] <- NA  
if(!identical(rownames(df), rownames(meta))) stop('Rowname issue')  
df$sample <- meta$sample  
df$cellState <- meta$RNA_state_abbr
```

```
In [7]: #max over PCA & PCA + Harmony for this feature  
cLISI_max <- 5.6  
iLISI_max <- 9.5  
sKNN_max <- 54  
lsKLD_max <- 4
```

```
In [8]: cLISI_med <- median(df$cLISI)  
iLISI_med <- median(df$iLISI)
```

```
sKNN_med <- median(df$sKNN)
lsKLD_med <- median(df$lsKLD)
```

```
In [9]: options(repr.plot.height=6,repr.plot.width=7.5)
g <- plot_fig(df,'UMAP1','UMAP2','cellState',method,feature,'RA T Cell\nnState',order='rand',
              plotColors=colors)
print(g)
if(!is.na(save_dir)){
    ggsave(file=paste(sep=' ',save_dir,outFile_prefix,'_ATAC_UMAP_cellState',file_extension),
           plot=g,units='in',height=6,width=7.5,dpi=300)
}

options(repr.plot.height=6,repr.plot.width=7.75)
g <- plot_fig(df,'UMAP1','UMAP2','sample',method,feature,'Sample',order='rand',
              plotColors=colors)
print(g)
if(!is.na(save_dir)){
    ggsave(file=paste(sep=' ',save_dir,outFile_prefix,'_ATAC_UMAP_sample',file_extension),
           plot=g,units='in',height=6,width=7.75,dpi=300)
}

options(repr.plot.height=6,repr.plot.width=6.75)
g <- plot_fig(df,'UMAP1','UMAP2','cLISI',method,feature,'cLISI',order='asc',
              plotColors=sep=',',good_color,midpt_color,cLISI_color,cLISI_med,1,cLISI_max)
print(g)
if(!is.na(save_dir)){
    ggsave(file=paste(sep=' ',save_dir,outFile_prefix,'_ATAC_UMAP_cLISI',file_extension),
           plot=g,units='in',height=6,width=6.75,dpi=300)
}

options(repr.plot.height=6,repr.plot.width=7)
g <- plot_fig(df,'UMAP1','UMAP2','iLISI',method,feature,'iLISI',order='desc',
              plotColors=sep=',',iLISI_color,midpt_color,good_color,iLISI_med,1,iLISI_max)
print(g)
if(!is.na(save_dir)){
    ggsave(file=paste(sep=' ',save_dir,outFile_prefix,'_ATAC_UMAP_iLISI',file_extension),
           plot=g,units='in',height=6,width=7,dpi=300)
}

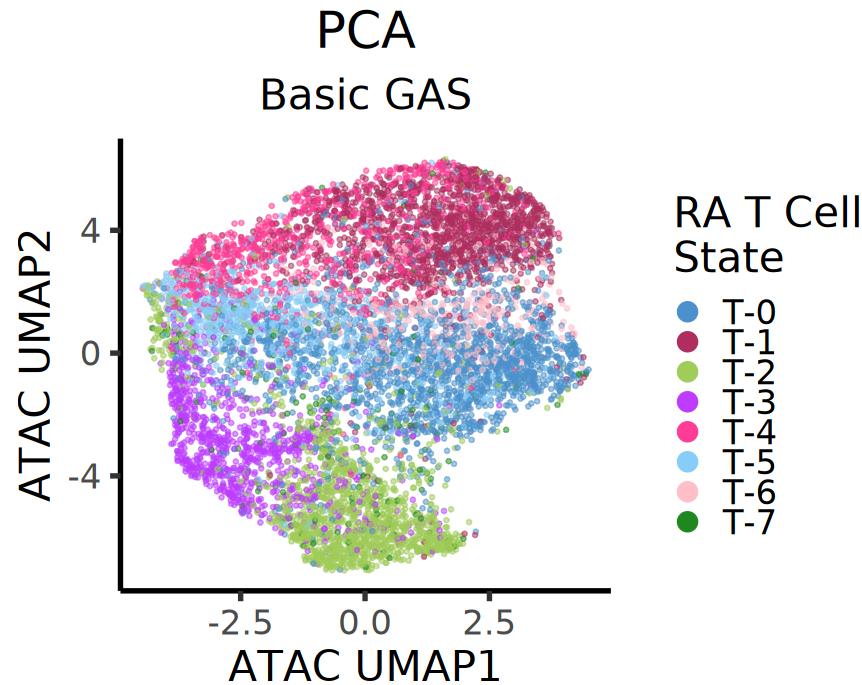
options(repr.plot.height=6,repr.plot.width=7.25)
g <- plot_fig(df,'UMAP1','UMAP2','sKNN_log10',method,feature,'Multiome\nnKNN\nnlog10\nnk=100',order='desc',
              NAcolor='#be91fe', #a shade darker than the lowest color
```

```

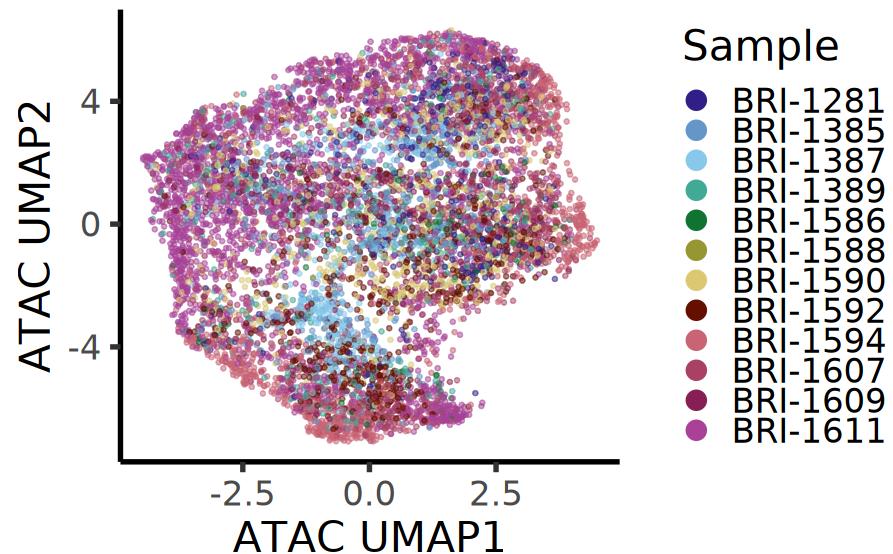
        plotColors=paste(sep=' ',sKNN_color,midpt_color,good_color,log10(sKNN_med),0,log10(sKNN_max))
print(g)
if(!is.na(save_dir)){
  ggsave(file= paste(sep=' ',save_dir,outFile_prefix,'_ATAC_UMAP_sKNN',file_extension),
         plot=g,units='in',height=6,width=7.25,dpi=300)
}

options(repr.plot.height=6,repr.plot.width=7.25)
g <- plot_fig(df,'UMAP1','UMAP2','lsKLD',method,feature,'lsKLD\nk=100',order='asc',
              plotColors= paste(sep=' ',good_color,midpt_color,lsKLD_color,lsKLD_med,0,lsKLD_max))
print(g)
if(!is.na(save_dir)){
  ggsave(file= paste(sep=' ',save_dir,outFile_prefix,'_ATAC_UMAP_lsKLD',file_extension),
         plot=g,units='in',height=6,width=7.25,dpi=300)
}

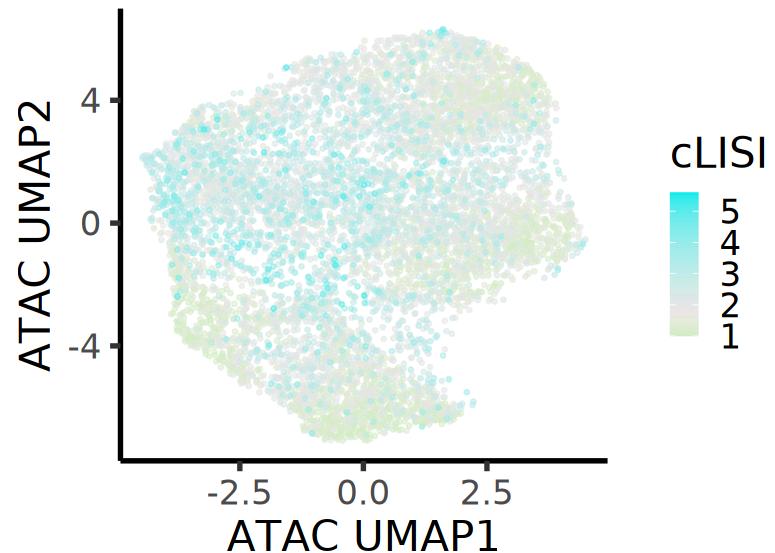
```



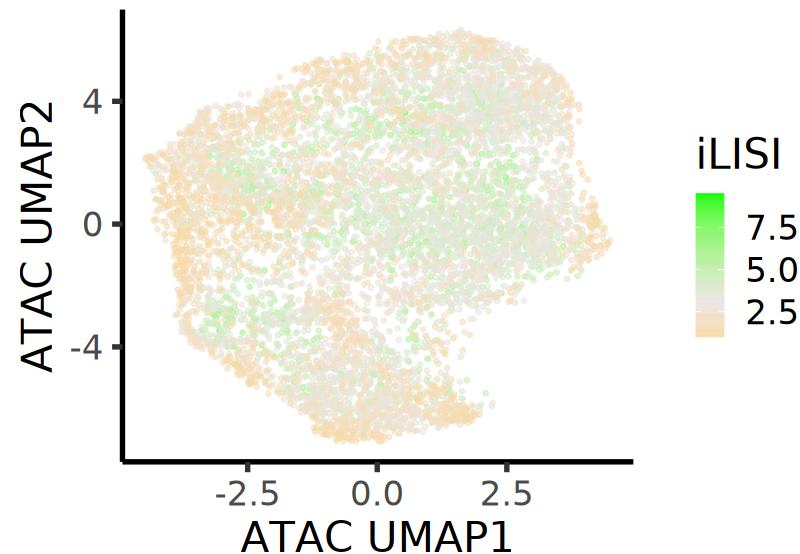
PCA  
Basic GAS



PCA  
Basic GAS

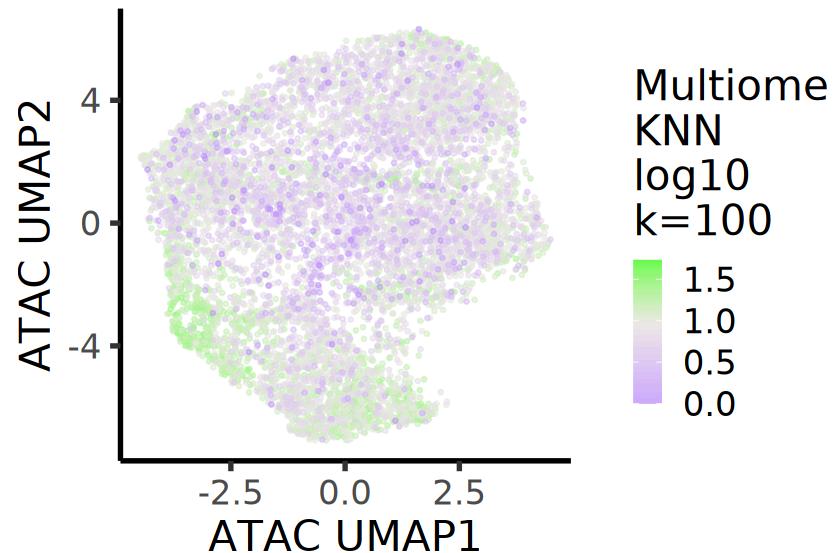


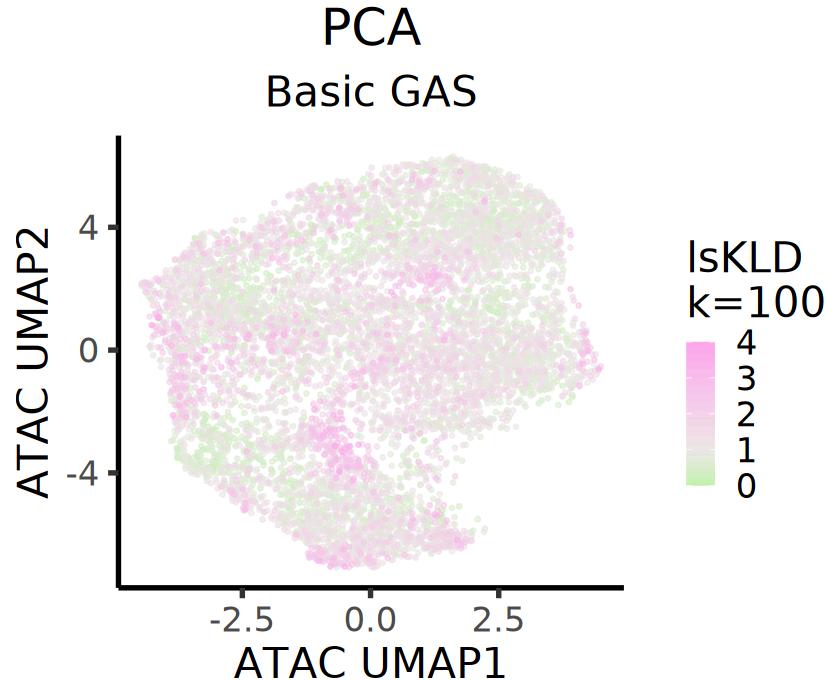
PCA  
Basic GAS



# PCA

## Basic GAS





```
In [10]: noHarmony_df <- df[sort(rownames(df)),]
```

## Basic GAS - PCA + Harmony

```
In [11]: method <- 'PCA + Harmony'
feature <- 'Basic GAS'
method_feature_prefix <- paste(sep='_', gsub(' ', '_', gsub(' \\\+ Harmony', '_Harmony', method)),
                                gsub(' ', '_', feature))
outFile_prefix <- paste(sep="_", dataset_str, method_feature_prefix)

df <- readRDS(paste(sep=' ', data_dir, dataset_str, '_pilotData_', method_feature_prefix, '.rds'))
df$sKNN_log10 <- log10(df$sKNN)
df[which(is.infinite(df$sKNN_log10)), 'sKNN_log10'] <- NA
if(!identical(rownames(df), rownames(meta))) stop('Rowname issue')
df$sample <- meta$sample
df$cellState <- meta$RNA_state_abbr
```

```
In [12]: options(repr.plot.height=6,repr.plot.width=7.5)
g <- plot_fig(df,'UMAP1','UMAP2','cellState',method,feature,'RA T Cell\nnState',order='rand',
              plotColors=colors)
print(g)
if(!is.na(save_dir)){
  ggsave(file=paste(sep='',save_dir,outFile_prefix,'_ATAC_UMAP_cellState',file_extension),
         plot=g,units='in',height=6,width=7.5,dpi=300)
}

options(repr.plot.height=6,repr.plot.width=7.75)
g <- plot_fig(df,'UMAP1','UMAP2','sample',method,feature,'Sample',order='rand',
              plotColors=colors)
print(g)
if(!is.na(save_dir)){
  ggsave(file=paste(sep='',save_dir,outFile_prefix,'_ATAC_UMAP_sample',file_extension),
         plot=g,units='in',height=6,width=7.75,dpi=300)
}

options(repr.plot.height=6,repr.plot.width=6.75)
g <- plot_fig(df,'UMAP1','UMAP2','cLISI',method,feature,'cLIST',order='asc',
              plotColors= paste(sep=',',good_color,midpt_color,cLISI_color,cLISI_med,1,cLISI_max))
print(g)
if(!is.na(save_dir)){
  ggsave(file=paste(sep='',save_dir,outFile_prefix,'_ATAC_UMAP_cLISI',file_extension),
         plot=g,units='in',height=6,width=6.75,dpi=300)
}

options(repr.plot.height=6,repr.plot.width=7)
g <- plot_fig(df,'UMAP1','UMAP2','iLISI',method,feature,'iLISI',order='desc',
              plotColors= paste(sep=',',iLISI_color,midpt_color,good_color,iLISI_med,1,iLISI_max))
print(g)
if(!is.na(save_dir)){
  ggsave(file=paste(sep='',save_dir,outFile_prefix,'_ATAC_UMAP_iLISI',file_extension),
         plot=g,units='in',height=6,width=7,dpi=300)
}

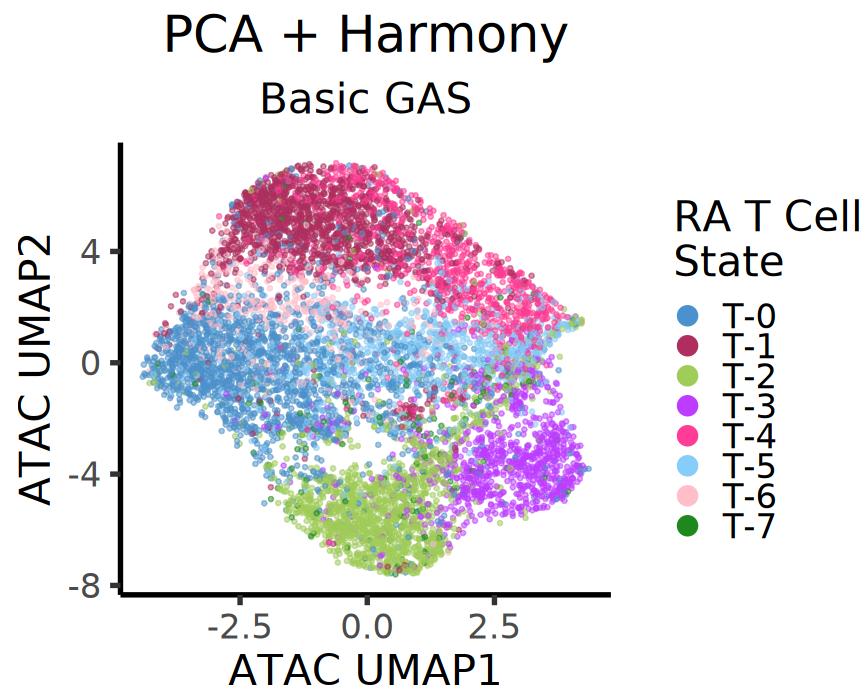
options(repr.plot.height=6,repr.plot.width=7.25)
g <- plot_fig(df,'UMAP1','UMAP2','sKNN_log10',method,feature,'Multiome\nnKNN\nnlog10\nnk=100',order='desc',
              NAcolor='#be91fe', #a shade darker than the lowest color
              plotColors= paste(sep=',',sKNN_color,midpt_color,good_color,log10(sKNN_med),0,log10(sKNN_max))
print(g)
```

```

if(!is.na(save_dir)){
  ggsave(file=paste(sep=' ', save_dir, outFile_prefix, '_ATAC_UMAP_sKNN', file_extension),
         plot=g, units='in', height=6, width=7.25, dpi=300)
}

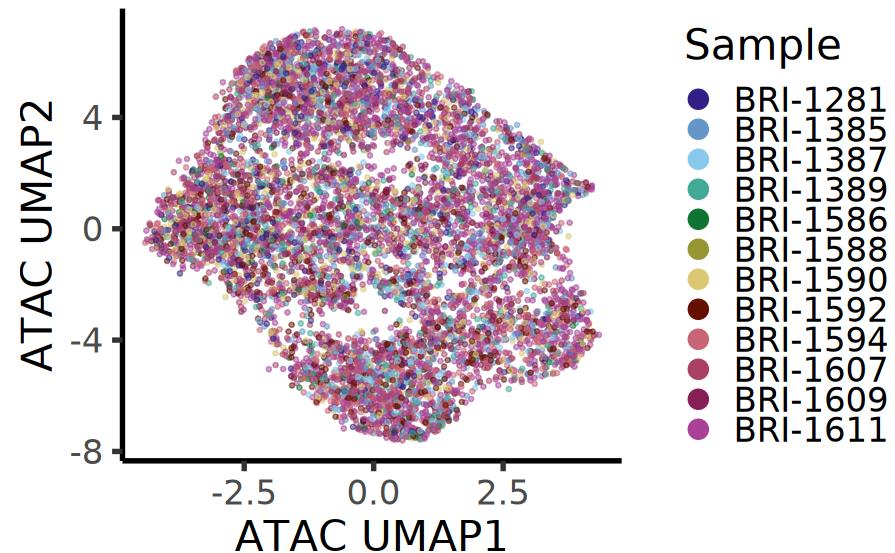
options(repr.plot.height=6, repr.plot.width=7.25)
g <- plot_fig(df,'UMAP1','UMAP2','lsKLD',method,feature,'lsKLD\nk=100',order='asc',
              plotColors= paste(sep=' ', good_color, midpt_color, lsKLD_color, lsKLD_med, 0, lsKLD_max))
print(g)
if(!is.na(save_dir)){
  ggsave(file=paste(sep=' ', save_dir, outFile_prefix, '_ATAC_UMAP_lsKLD', file_extension),
         plot=g, units='in', height=6, width=7.25, dpi=300)
}

```



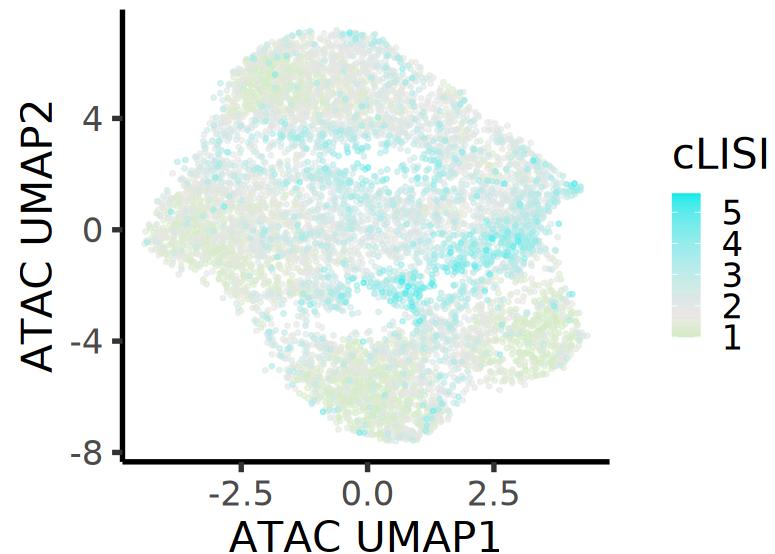
# PCA + Harmony

Basic GAS



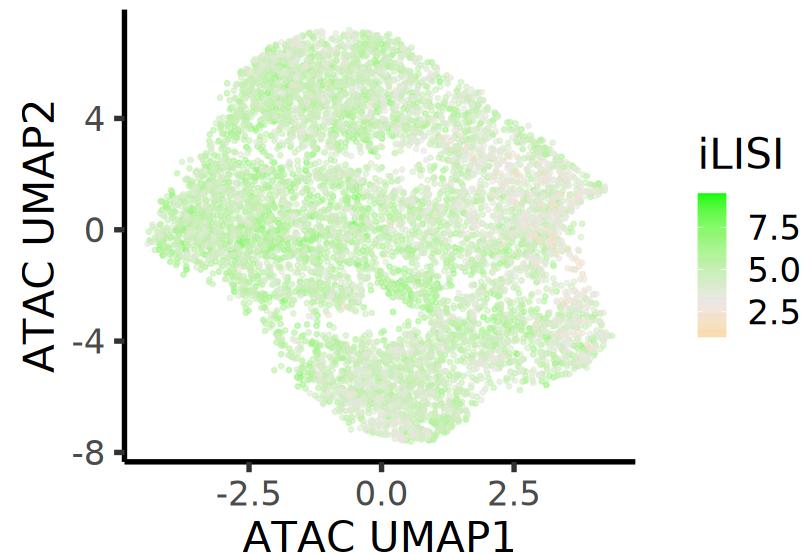
# PCA + Harmony

Basic GAS



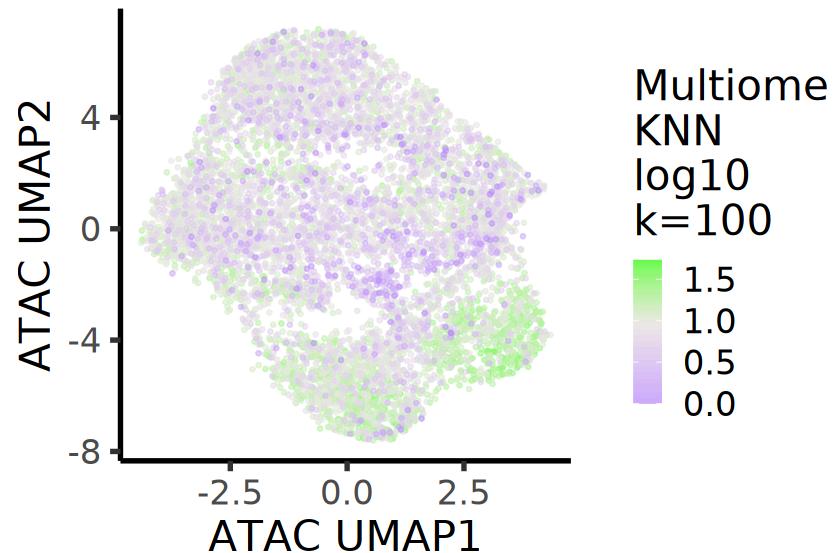
# PCA + Harmony

Basic GAS



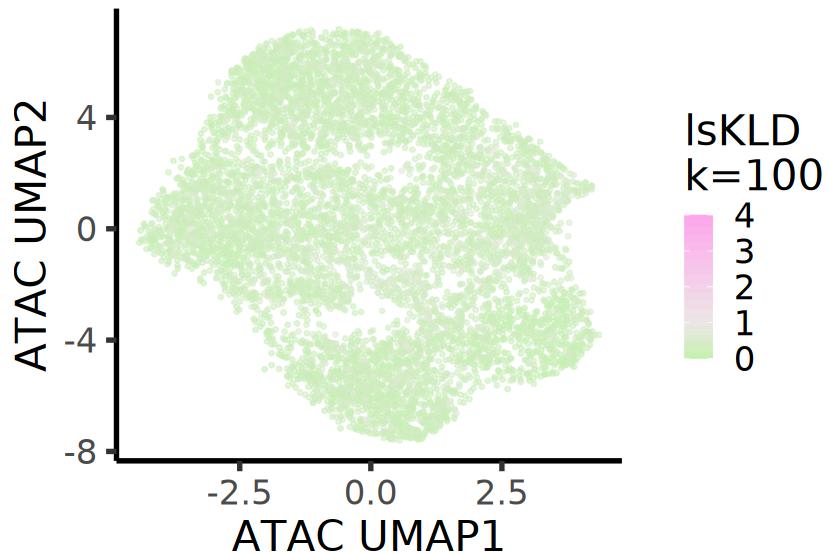
# PCA + Harmony

## Basic GAS



## PCA + Harmony

### Basic GAS



```
In [13]: harmony_df <- df[sort(rownames(df)),]
```

### Basic GAS - Compare Metrics

```
In [14]: identical(rownames(noHarmony_df), rownames(harmony_df))
```

TRUE

```
In [15]: toPlot <- noHarmony_df[,c('cLISI','iLISI','sKNN','lsKLD')]
colnames(toPlot) <- c('PCA_cLISI','PCA_iLISI','PCA_sKNN','PCA_lsKLD')

toPlot <- cbind(toPlot,harmony_df[,c('cLISI','iLISI','sKNN','lsKLD')])
colnames(toPlot)[5:8] <- c('Harmony_cLISI','Harmony_iLISI','Harmony_sKNN','Harmony_lsKLD')

toPlot$cell <- rownames(toPlot)
toPlot$sample <- noHarmony_df$sample

toPlot <- gather(toPlot,'combo','value',all_of(colnames(toPlot)[1:8]))
```

```

splits <- str_split_fixed(toPlot$combo, '_', 2)
toPlot$method <- splits[,1]
toPlot$metric <- splits[,2]

toPlot$method <- factor(toPlot$method, levels=c('PCA','Harmony'))
toPlot$metric <- factor(toPlot$metric, levels=c('cLISI','sKNN','iLISI','lsKLD'))

options(repr.plot.height=6,repr.plot.width=20)
g <- ggplot(toPlot,aes(x=method,y=value,color=metric)) + geom_boxplot(lwd = 1) +
  facet_wrap(~metric,scales='free',nrow=1,ncol=4,labeller=labeller(metric = c('sKNN'='Multiome KNN'),
  theme_bw(base_size=25) + theme(legend.position="none") +
  labs(x='Method',y='Metric Value',title=CT_str,subtitle='Basic GAS') +
  theme(plot.title = element_text(hjust = 0.5),plot.subtitle = element_text(hjust = 0.5)) +
  scale_color_manual(values=c('cLISI'=cLISI_color,'iLISI'=iLISI_color,
  'sKNN'=sKNN_color,'lsKLD'=lsKLD_color)) +
  scale_x_discrete(labels=c('PCA', 'PCA + Harmony')))

stat.test <- toPlot %>% group_by(metric) %>% t_test(value ~ method, paired=TRUE) %>%
  add_significance() %>% add_xy_position(x = "method")
stat.test

g <- g + stat_pvalue_manual(stat.test)

print(g)

if(!is.na(save_dir)){
  outFile_prefix <- paste(sep="_",dataset_str,gsub(' ','_',feature))
  ggsave(file=paste(sep='',save_dir,outFile_prefix,'_metrics',file_extension),
  plot=g,units='in',height=6,width=20,dpi=300)
}

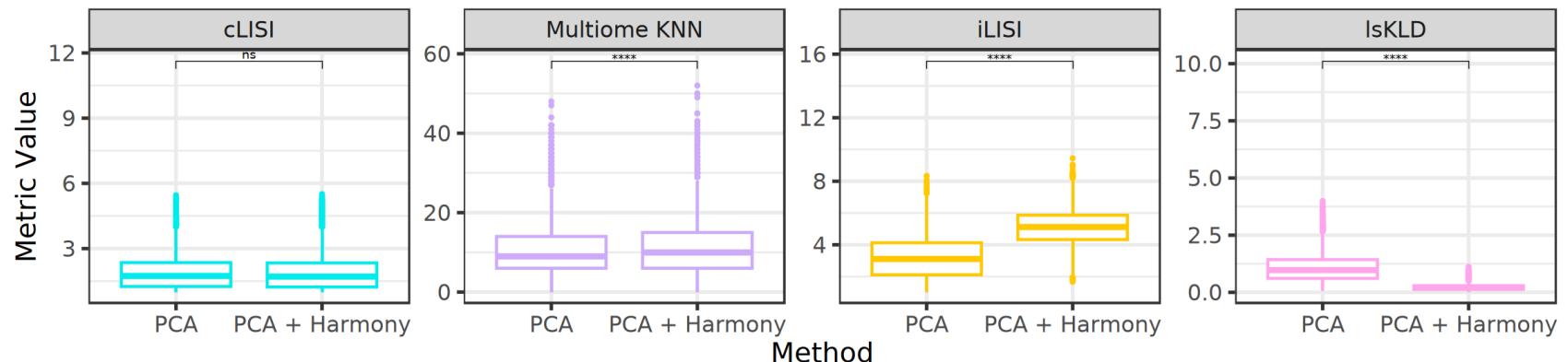
```

A rstatix\_test: 4 × 14

metric	.y.	group1	group2	n1	n2	statistic	df	p	p.signif	y.position	groups	NA	
	<fct>	<chr>	<chr>	<int>	<int>	<dbl>	<dbl>	<dbl>	<chr>	<dbl>	<named list>	<dbl>	<chr>
cLISI	value	PCA	Harmony	8078	8078	1.370333	8077	1.71e-01	ns	11.61572	PCA	PCA	
sKNN	value	PCA	Harmony	8078	8078	-14.640267	8077	6.37e-48	****	58.10572	Harmony	Harmony	Harmony
iLISI	value	PCA	Harmony	8078	8078	-106.499169	8077	0.00e+00	****	15.55172	PCA	PCA	
IsKLD	value	PCA	Harmony	8078	8078	125.271224	8077	0.00e+00	****	10.10272	Harmony	Harmony	Harmony

### RA T Cell State

#### Basic GAS



In [16]: #confirming earlier max values okay!

```
max(toPlot[which(toPlot$metric=='cLISI'),'value'])
max(toPlot[which(toPlot$metric=='iLISI'),'value'])
max(toPlot[which(toPlot$metric=='sKNN'),'value'])
max(toPlot[which(toPlot$metric=='lsKLD'),'value'])
```

5.51032327685906

9.44562329126135

52

3.99689527792075

## Peaks - PCA

```
In [17]: method <- 'PCA'
feature <- 'Peaks'
method_feature_prefix <- paste(sep='_', gsub(' ', '', gsub(' \\\+ Harmony', '_Harmony', method)), 
                                gsub(' ', '', feature))
outFile_prefix <- paste(sep="_", dataset_str, method_feature_prefix)

df <- readRDS(paste(sep='', data_dir, dataset_str, '_pilotData_', method_feature_prefix, '.rds'))
df$sKNN_log10 <- log10(df$sKNN)
df[which(is.infinite(df$sKNN_log10)), 'sKNN_log10'] <- NA
if(!identical(rownames(df), rownames(meta))) stop('Rowname issue')
df$sample <- meta$sample
df$cellState <- meta$RNA_state_abbr
```

```
In [18]: #max over PCA & PCA + Harmony for this feature
cLISI_max <- 5.7
iLISI_max <- 8.8
sKNN_max <- 57
lsKLD_max <- 5.6
```

```
In [19]: cLISI_med <- median(df$cLISI)
iLISI_med <- median(df$iLISI)
sKNN_med <- median(df$sKNN)
lsKLD_med <- median(df$lsKLD)
```

```
In [20]: options(repr.plot.height=6, repr.plot.width=7.5)
g <- plot_fig(df, 'UMAP1', 'UMAP2', 'cellState', method, feature, 'RA T Cell\nnState', order='rand',
               plotColors=colors)
print(g)
if(!is.na(save_dir)){
    ggsave(file=paste(sep='', save_dir, outFile_prefix, '_ATAC_UMAP_cellState', file_extension),
           plot=g, units='in', height=6, width=7.5, dpi=300)
}

options(repr.plot.height=6, repr.plot.width=7.75)
g <- plot_fig(df, 'UMAP1', 'UMAP2', 'sample', method, feature, 'Sample', order='rand',
               plotColors=colors)
print(g)
```

```

if(!is.na(save_dir)){
  ggsave(file=paste(sep=' ', save_dir,outFile_prefix,'_ATAC_UMAP_sample',file_extension),
         plot=g,units='in',height=6,width=7.75,dpi=300)
}

options(repr.plot.height=6,repr.plot.width=6.75)
g <- plot_fig(df,'UMAP1','UMAP2','cLISI',method,feature,'cLISI',order='asc',
              plotColors= paste(sep=' ',good_color,midpt_color,cLISI_color,cLISI_med,1,cLISI_max))
print(g)
if(!is.na(save_dir)){
  ggsave(file=paste(sep=' ', save_dir,outFile_prefix,'_ATAC_UMAP_cLISI',file_extension),
         plot=g,units='in',height=6,width=6.75,dpi=300)
}

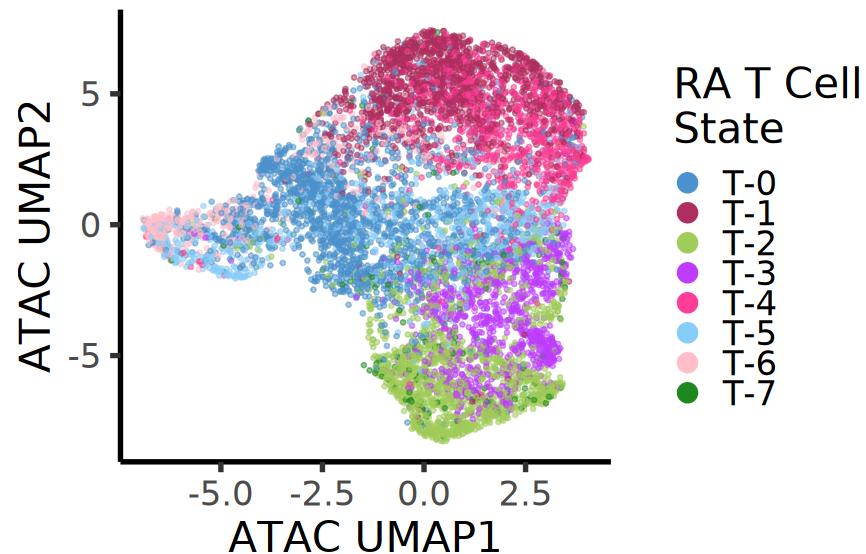
options(repr.plot.height=6,repr.plot.width=7)
g <- plot_fig(df,'UMAP1','UMAP2','iLISI',method,feature,'iLISI',order='desc',
              plotColors= paste(sep=' ',iLISI_color,midpt_color,good_color,iLISI_med,1,iLISI_max))
print(g)
if(!is.na(save_dir)){
  ggsave(file=paste(sep=' ', save_dir,outFile_prefix,'_ATAC_UMAP_iLISI',file_extension),
         plot=g,units='in',height=6,width=7,dpi=300)
}

options(repr.plot.height=6,repr.plot.width=7.25)
g <- plot_fig(df,'UMAP1','UMAP2','sKNN_log10',method,feature,'Multiome\nKNN\nlog10\nnk=100',order='desc',
              NAcolor='#be91fe', #a shade darker than the lowest color
              plotColors= paste(sep=' ',sKNN_color,midpt_color,good_color,log10(sKNN_med),0,log10(sKNN_max))
print(g)
if(!is.na(save_dir)){
  ggsave(file=paste(sep=' ', save_dir,outFile_prefix,'_ATAC_UMAP_sKNN',file_extension),
         plot=g,units='in',height=6,width=7.25,dpi=300)
}

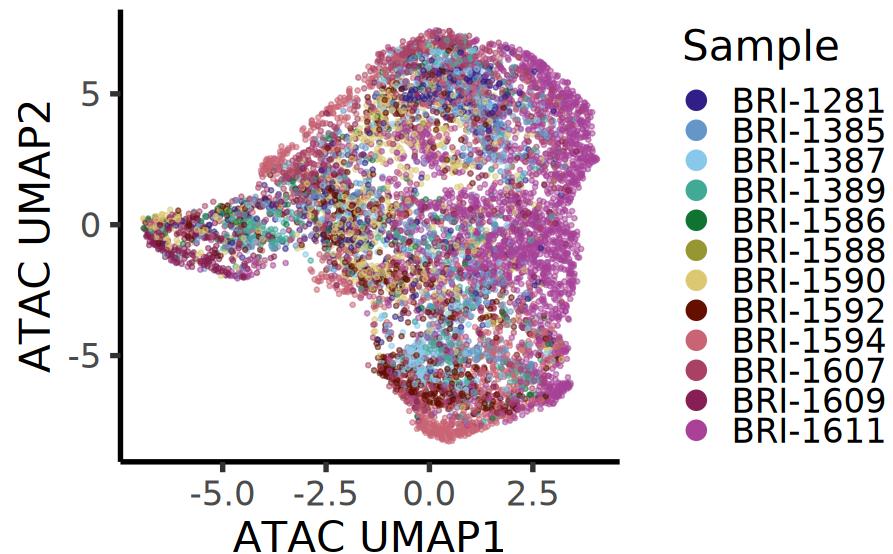
options(repr.plot.height=6,repr.plot.width=7.25)
g <- plot_fig(df,'UMAP1','UMAP2','lsKLD',method,feature,'lsKLD\nnk=100',order='asc',
              plotColors= paste(sep=' ',good_color,midpt_color,lsKLD_color,lsKLD_med,0,lsKLD_max))
print(g)
if(!is.na(save_dir)){
  ggsave(file=paste(sep=' ', save_dir,outFile_prefix,'_ATAC_UMAP_lsKLD',file_extension),
         plot=g,units='in',height=6,width=7.25,dpi=300)
}

```

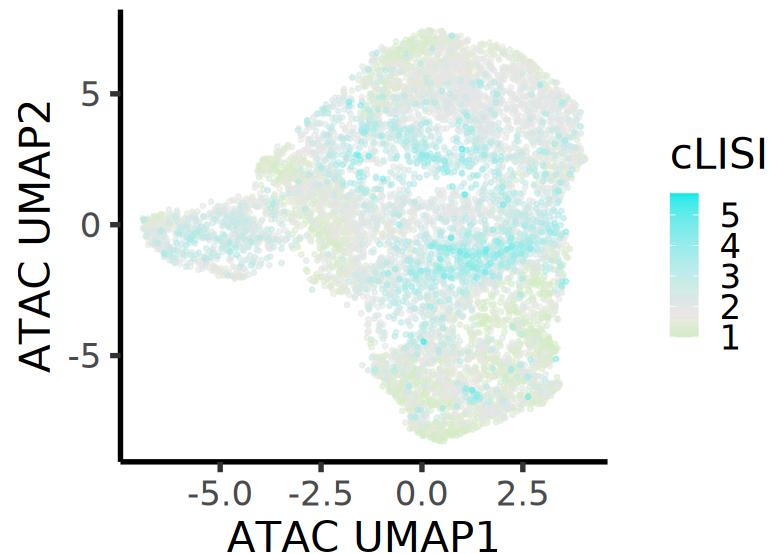
# PCA Peaks



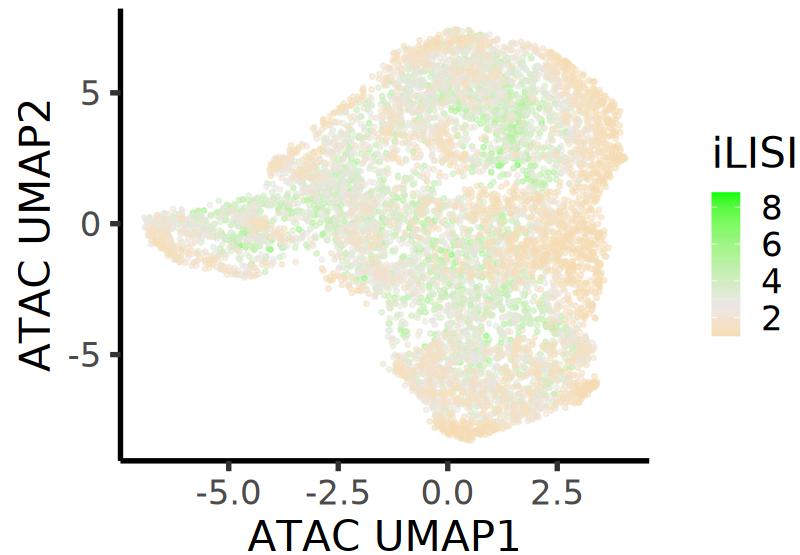
# PCA Peaks



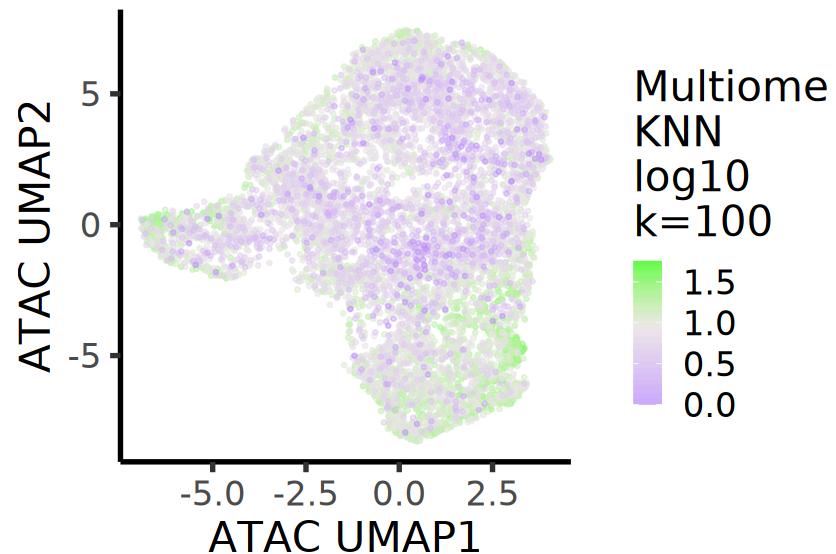
# PCA Peaks

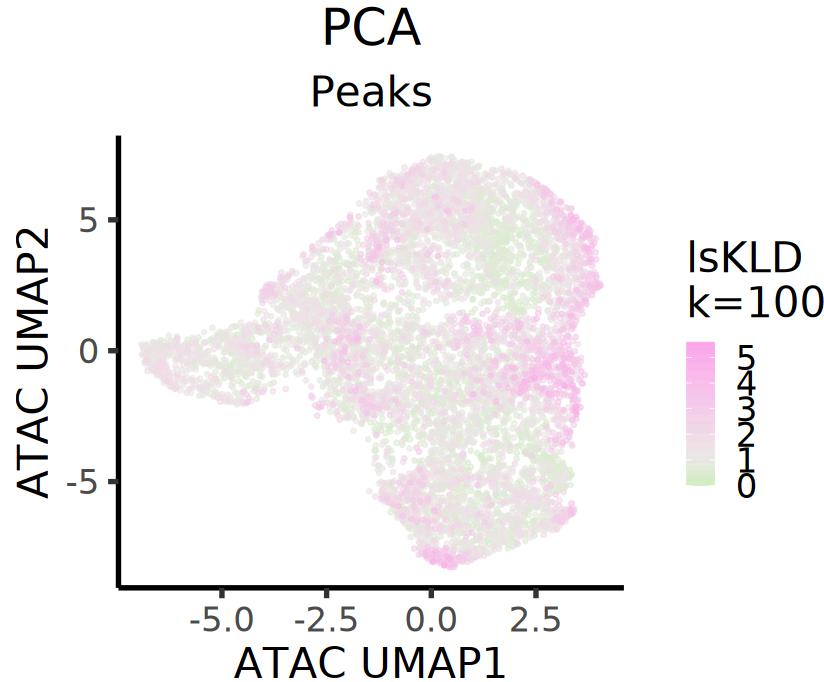


# PCA Peaks



# PCA Peaks





```
In [21]: noHarmony_df <- df[sort(rownames(df)),]
```

## Peaks - PCA + Harmony

```
In [22]: method <- 'PCA + Harmony'
feature <- 'Peaks'
method_feature_prefix <- paste(sep='_', gsub(' ', '', gsub(' \\\+ Harmony', '_Harmony', method)),
                                gsub(' ', '', feature))
outFile_prefix <- paste(sep="_", dataset_str, method_feature_prefix)

df <- readRDS(paste(sep=' ', data_dir, dataset_str, '_pilotData_', method_feature_prefix, '.rds'))
df$sKNN_log10 <- log10(df$sKNN)
df[which(is.infinite(df$sKNN_log10)), 'sKNN_log10'] <- NA
if(!identical(rownames(df), rownames(meta))) stop('Rowname issue')
df$sample <- meta$sample
df$cellState <- meta$RNA_state_abbr
```

In [23]:

```
options(repr.plot.height=6,repr.plot.width=7.5)
g <- plot_fig(df,'UMAP1','UMAP2','cellState',method,feature,'RA T Cell\nnState',order='rand',
              plotColors=colors)
print(g)
if(!is.na(save_dir)){
  ggsave(file=paste(sep='',save_dir,outFile_prefix,'_ATAC_UMAP_cellState',file_extension),
         plot=g,units='in',height=6,width=7.5,dpi=300)
}

options(repr.plot.height=6,repr.plot.width=7.75)
g <- plot_fig(df,'UMAP1','UMAP2','sample',method,feature,'Sample',order='rand',
              plotColors=colors)
print(g)
if(!is.na(save_dir)){
  ggsave(file=paste(sep='',save_dir,outFile_prefix,'_ATAC_UMAP_sample',file_extension),
         plot=g,units='in',height=6,width=7.75,dpi=300)
}

options(repr.plot.height=6,repr.plot.width=6.75)
g <- plot_fig(df,'UMAP1','UMAP2','cLISI',method,feature,'cLIST',order='asc',
              plotColors=sep=',',good_color,midpt_color,cLISI_color,cLISI_med,1,cLISI_max)
print(g)
if(!is.na(save_dir)){
  ggsave(file=paste(sep='',save_dir,outFile_prefix,'_ATAC_UMAP_cLISI',file_extension),
         plot=g,units='in',height=6,width=6.75,dpi=300)
}

options(repr.plot.height=6,repr.plot.width=7)
g <- plot_fig(df,'UMAP1','UMAP2','iLISI',method,feature,'iLISI',order='desc',
              plotColors=sep=',',iLISI_color,midpt_color,good_color,iLISI_med,1,iLISI_max)
print(g)
if(!is.na(save_dir)){
  ggsave(file=paste(sep='',save_dir,outFile_prefix,'_ATAC_UMAP_iLISI',file_extension),
         plot=g,units='in',height=6,width=7,dpi=300)
}

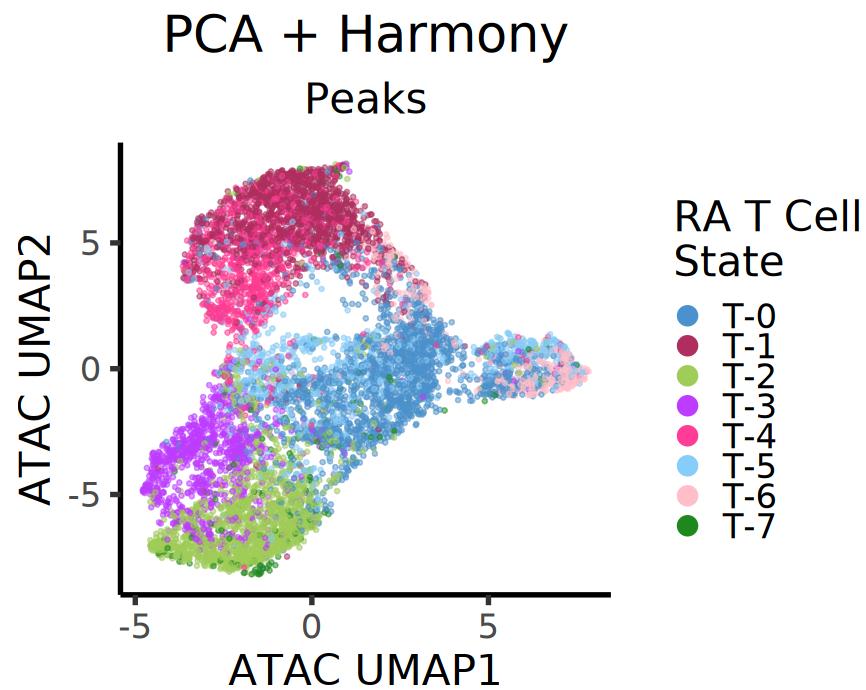
options(repr.plot.height=6,repr.plot.width=7.25)
g <- plot_fig(df,'UMAP1','UMAP2','sKNN_log10',method,feature,'Multiome\nnKNN\nnlog10\nnk=100',order='desc',
              NAcolor='#be91fe', #a shade darker than the lowest color
              plotColors=sep=',',sKNN_color,midpt_color,good_color,log10(sKNN_med),0,log10(sKNN_max))
print(g)
```

```

if(!is.na(save_dir)){
  ggsave(file=paste(sep=' ', save_dir, outFile_prefix, '_ATAC_UMAP_sKNN', file_extension),
         plot=g, units='in', height=6, width=7.25, dpi=300)
}

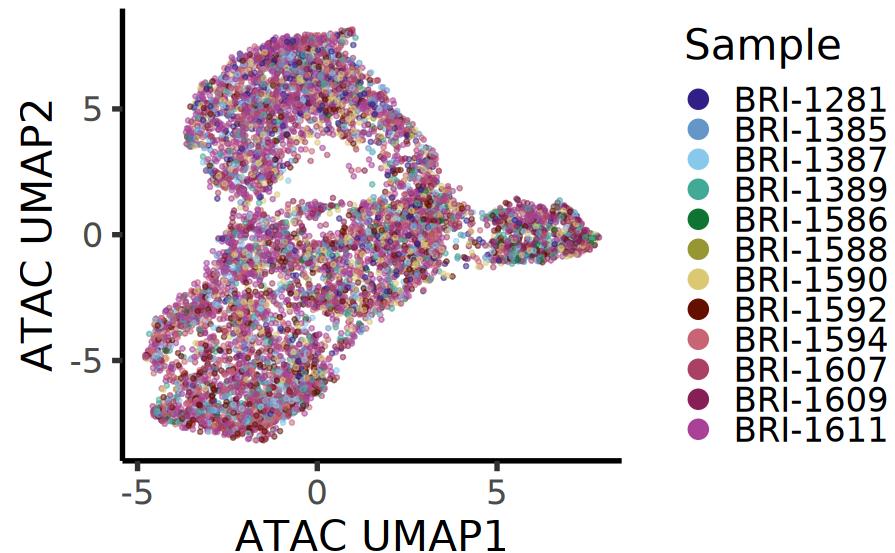
options(repr.plot.height=6, repr.plot.width=7.25)
g <- plot_fig(df, 'UMAP1', 'UMAP2', 'lsKLD', method, feature, 'lsKLD\nnk=100', order='asc',
              plotColors= paste(sep=' ', good_color, midpt_color, lsKLD_color, lsKLD_med, 0, lsKLD_max))
print(g)
if(!is.na(save_dir)){
  ggsave(file=paste(sep=' ', save_dir, outFile_prefix, '_ATAC_UMAP_lsKLD', file_extension),
         plot=g, units='in', height=6, width=7.25, dpi=300)
}

```



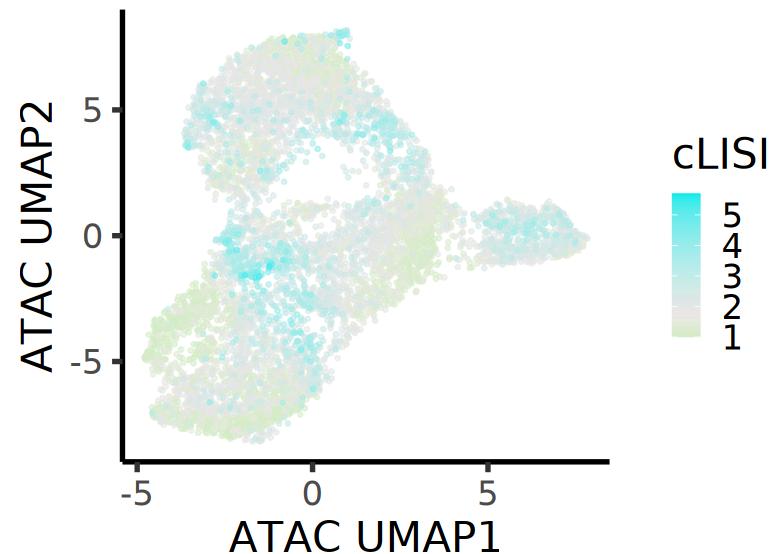
# PCA + Harmony

## Peaks



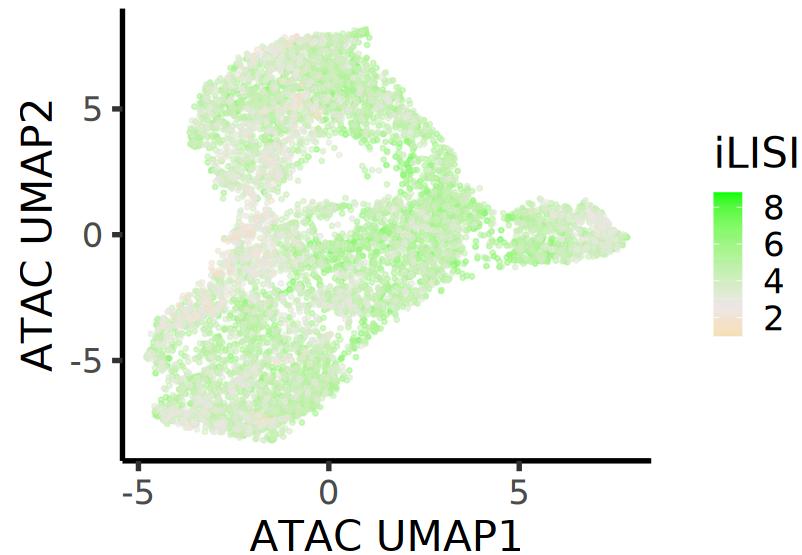
# PCA + Harmony

Peaks



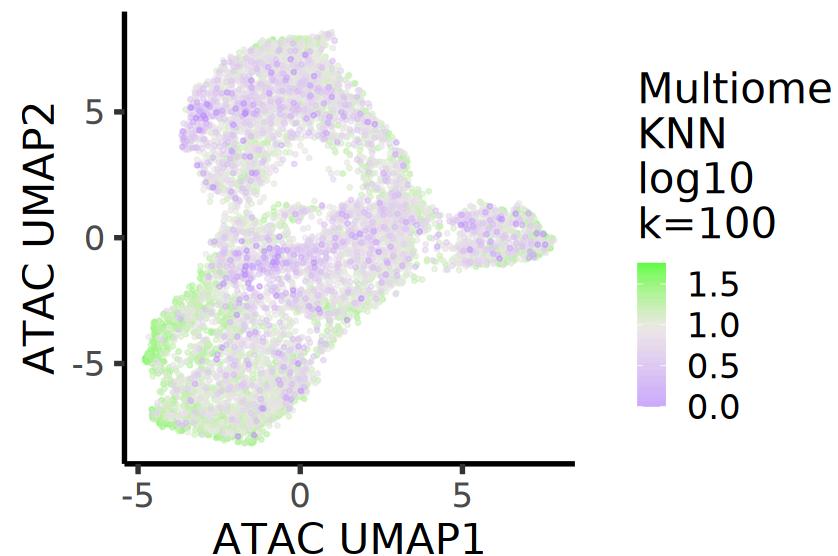
# PCA + Harmony

Peaks

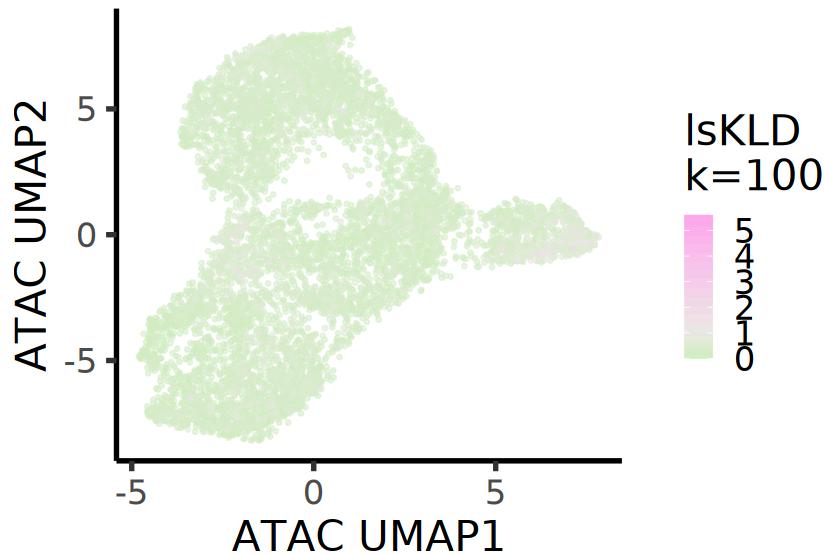


# PCA + Harmony

## Peaks



## PCA + Harmony Peaks



```
In [24]: harmony_df <- df[sort(rownames(df)),]
```

## Peaks - Compare Metrics

```
In [25]: identical(rownames(noHarmony_df), rownames(harmony_df))
```

TRUE

```
In [26]: toPlot <- noHarmony_df[,c('cLISI','iLISI','sKNN','lS KLD')]
colnames(toPlot) <- c('PCA_cLISI','PCA_iLISI','PCA_sKNN','PCA_lS KLD')

toPlot <- cbind(toPlot,harmony_df[,c('cLISI','iLISI','sKNN','lS KLD')])
colnames(toPlot)[5:8] <- c('Harmony_cLISI','Harmony_iLISI','Harmony_sKNN','Harmony_lS KLD')

toPlot$cell <- rownames(toPlot)
toPlot$sample <- noHarmony_df$sample

toPlot <- gather(toPlot,'combo','value',all_of(colnames(toPlot)[1:8]))
```

```

splits <- str_split_fixed(toPlot$combo,' ',2)
toPlot$method <- splits[,1]
toPlot$metric <- splits[,2]

toPlot$method <- factor(toPlot$method,levels=c('PCA','Harmony'))
toPlot$metric <- factor(toPlot$metric,levels=c('cLISI','sKNN','iLISI','lsKLD'))

options(repr.plot.height=6,repr.plot.width=20)
g <- ggplot(toPlot,aes(x=method,y=value,color=metric)) + geom_boxplot(lwd = 1) +
  facet_wrap(~metric,scales='free',nrow=1,ncol=4,labeller=labeller(metric = c('sKNN'='Multiome KNN'),
  theme_bw(base_size=25) + theme(legend.position="none") +
  labs(x='Method',y='Metric Value',title=CT_str,subtitle='Peaks') +
  theme(plot.title = element_text(hjust = 0.5),plot.subtitle = element_text(hjust = 0.5)) +
  scale_color_manual(values=c('cLISI'=cLISI_color,'iLISI'=iLISI_color,
  'sKNN'=sKNN_color,'lsKLD'=lsKLD_color)) +
  scale_x_discrete(labels=c('PCA'='PCA', 'Harmony'='PCA + Harmony')))

stat.test <- toPlot %>% group_by(metric) %>% t_test(value ~ method, paired=TRUE) %>%
  add_significance() %>% add_xy_position(x = "method")
stat.test

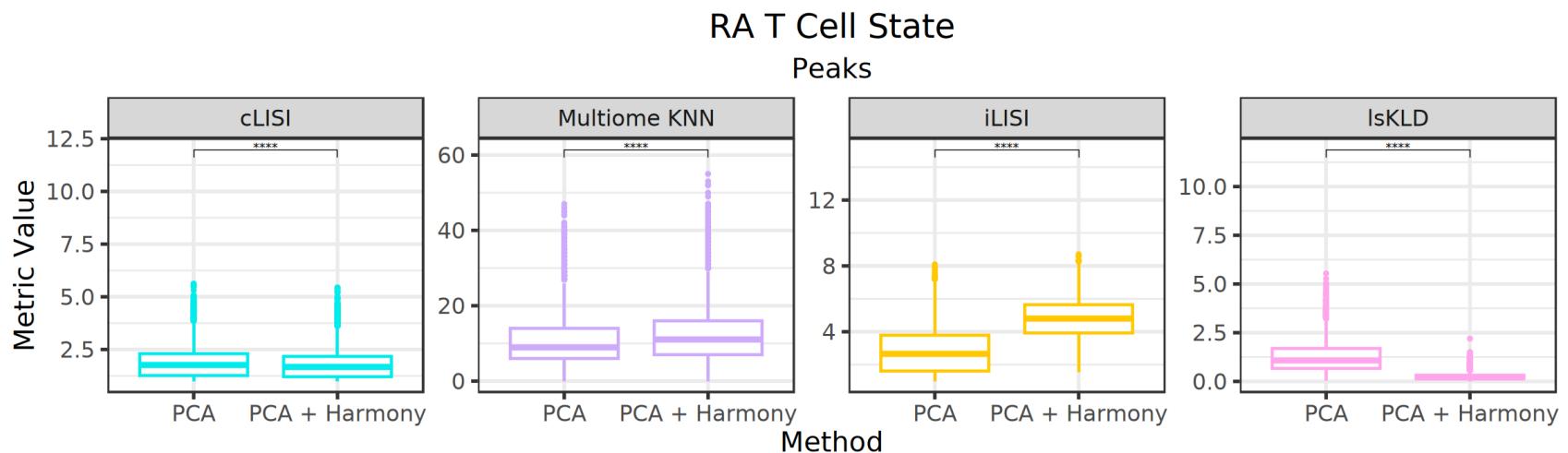
g <- g + stat_pvalue_manual(stat.test)

print(g)

if(!is.na(save_dir)){
  outFile_prefix <- paste(sep="_",dataset_str,gsub(' ','_',feature))
  ggsave(file=paste(sep='',save_dir,outFile_prefix,'_metrics',file_extension),
  plot=g,units='in',height=6,width=20,dpi=300)
}

```

A rstatix_test: 4 × 14																
metric	.y.	group1	group2	n1	n2	statistic	df	p	p.signif	y.position	groups	NA				
<fct>	<chr>	<chr>	<chr>	<int>	<int>	<dbl>	<dbl>	<dbl>	<chr>	<dbl>	<named list>	<dbl>	<dbl>	<dbl>	<dbl>	
cLISI	value	PCA	Harmony	8078	8078	13.56821	8077	1.76e-41	****	11.9666	PCA	PCA	P			
sKNN	value	PCA	Harmony	8078	8078	-34.56983	8077	2.39e-244	****	61.3366	Harmony	Harmony	Harmon			
iLISI	value	PCA	Harmony	8078	8078	-103.63667	8077	0.00e+00	****	15.0436	PCA	PCA	P			
lsKLD	value	PCA	Harmony	8078	8078	112.65567	8077	0.00e+00	****	11.8786	Harmony	Harmony	Harmon			



```
In [27]: #confirming earlier max values okay!
max(toPlot[which(toPlot$metric=='cLISI'),'value'])
max(toPlot[which(toPlot$metric=='iLISI'),'value'])
max(toPlot[which(toPlot$metric=='sKNN'),'value'])
max(toPlot[which(toPlot$metric=='lsKLD'),'value'])
```

5.6297536855862

8.70696109466622

55

5.54245246622513

## Session Info

```
In [29]: sessionInfo()
```

```
R version 4.3.3 (2024-02-29)
Platform: x86_64-conda-linux-gnu (64-bit)
Running under: CentOS Linux 7 (Core)

Matrix products: default
BLAS/LAPACK: /PHShome/kew47/miniconda3/envs/integrateATAC/lib/libopenblas-r0.3.27.so; LAPACK version 3.1
2.0

locale:
[1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
[3] LC_TIME=en_US.UTF-8       LC_COLLATE=en_US.UTF-8
[5] LC_MONETARY=en_US.UTF-8   LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=en_US.UTF-8     LC_NAME=C
[9] LC_ADDRESS=C              LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

time zone: America/New_York
tzcode source: system (glibc)

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

other attached packages:
[1] repr_1.1.7   ggpubr_0.6.0  scales_1.3.0  ggrastr_1.0.2 ggplot2_3.5.1
[6] rstatix_0.7.2 stringr_1.5.1  tidyverse_1.3.1

loaded via a namespace (and not attached):
[1] utf8_1.2.4      generics_0.1.3   stringi_1.8.4   digest_0.6.37
[5] magrittr_2.0.3   evaluate_0.24.0  grid_4.3.3      pbdZMQ_0.3-11
[9] fastmap_1.2.0   jsonlite_1.8.8   backports_1.5.0  purrr_1.0.2
[13] fansi_1.0.6    textshaping_0.4.0 abind_1.4-5    cli_3.6.3
[17] rlang_1.1.4    crayon_1.5.3    munsell_0.5.1   base64enc_0.1-3
[21] withr_3.0.1    ggbeeswarm_0.7.2 tools_4.3.3    uuid_1.2-1
[25] ggsignif_0.6.4  dplyr_1.1.4    colorspace_2.1-1 broom_1.0.6
[29] IRdisplay_1.1   vctrs_0.6.5    R6_2.5.1       lifecycle_1.0.4
[33] car_3.1-2      viper_0.4.7    ragg_1.3.2     pkgconfig_2.0.3
[37] beeswarm_0.4.0  pillar_1.9.0   gtable_0.3.5   glue_1.7.0
[41] systemfonts_1.1.0 tibble_3.2.1   tidyselect_1.2.1 IRkernel_1.3.2
[45] farver_2.1.2    htmltools_0.5.8.1 carData_3.0-5 labeling_0.4.3
[49] Cairo_1.6-2    compiler_4.3.3
```

In [ ]: