

Plot Generation, per Figure

Load Libraries

```
library(Seurat)
library(scales)
library(viridis)
library(enrichR)
library(tidyselect)
library(ggpubr)
library(tidyverse)
library(SeuratWrappers)
library(purrr)
library(ggplot2)
library(cowplot)
library(grid)
library(gridExtra)
library(gtable)
library(reshape2)
library(circlize)
library(tidyverse)
library(Biobase)
library(gginnards)
library(dplyr)
library(igraph)
library(sctransform)
library(glmGamPoi)
library(UpSetR)
library(ComplexHeatmap)
library(ggridges)
library(patchwork)
library(knitr)
library(monocle3)
library(viridisLite)
```

Global Functions

```
# Loads an RData file, and returns it
loadRData <- function(fileName){
  load(fileName)
  get(ls()[ls() != "fileName"])
}

# Negates %in% operator
`%!in%` = Negate(`%in%`)
```

Object Nomenclature

```
AllCellTypes_Object # Integrated seurat object with all cell types
Epi_Object          # Integrated seurat object - epithelial cells
Epi_Object_Merged   # Merged seurat object - epithelial cells
Epi_CDS             # Monocle3 object converted from merged seurat object - epithelial cells
EoE_GERD_Epi_Object # Integrated seurat object - epithelial cells from HC, EoE, and GERD patients
```

Functions imported from Plot_Functions script

```
StackedVlnPlot           # Generate stacked violin plots from a list of genes
Proportion_Barplot_withMultipleComparisons # Generate bar plots testing differences in proportions, with multiple comparisons
Expression_BoxPlot        # Generate box plots to examine the distribution of continuous variables across cells, with multiple comparison
```

Figure 2

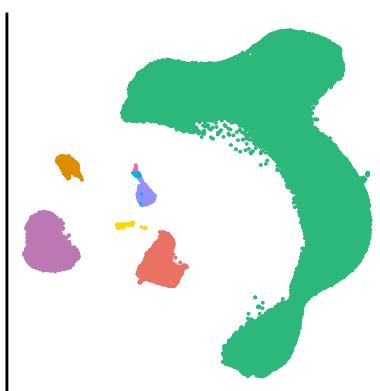
UMAP of Cell Types

```

# Assign Colors
coloring <- c("#2CB77B", "#Epithelial cells"
             "#BC78B2", "#T/NK cells"
             "#E97265", "#MNPs"
             "#DE8C00", "#Mast cells"
             "#FFD700", # B Cells
             "#9590FF", "#Endothelial cells"
             "#00B4FO", "#Fibroblasts"
             "#FF64B0") #Smooth muscle"

# Construct UMAP plot
left_join(data.frame(AllCellTypes_Object@reductions$umap@cell.embeddings) %>% rownames_to_column(var = "Cells"),
          data.frame(AllCellTypes_Object@meta.data %>% select(CellTypes)) %>% rownames_to_column(var = "Cells")) %>%
  ggplot(aes(x = UMAP_1, y = UMAP_2, color = CellTypes)) +
  geom_point(size = 0.01) +
  scale_color_manual(values = coloring,
                      name = "Cell Types") +
  theme(axis.line = element_line(colour = "black"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank(),
        plot.margin = unit(c(0, 0, 0, 0), "cm"),
        axis.text.y = element_blank(),
        axis.ticks.x = element_blank(),
        axis.ticks.y = element_blank(),
        axis.text.x = element_blank(),
        aspect.ratio = 1,
        legend.title = element_text(size = 7),
        legend.text = element_text(size = 7, lineheight = 0.2),
        legend.key = element_rect(size = 13,
                                   fill = NA)) +
  xlab("") + ylab("") +
  guides(color = guide_legend(override.aes = list(size = 3))) +
  ggtitle("")

```



Violin Plot of Cell Type Markers

```

# Assign cell type markers
markers <- c(# Epithelial Cells
             "KRT6A",
             "DSG3",
             # CD8 T cells
             "CD3D",
             #NK cells
             "NKG7",
             # Macrophages / Dendritic Cells / Monocytes
             "CD68",
             "CD207",
             "CD14",
             # Mast Cells
             "KIT",
             "CPA3",
             #B Cells
             "CD79A",
             "IGHA1",
             # Endothelial Cells
             "VWF",
             "CDH5",
             # Fibroblasts
             "DCN",

```

```

"COL1A1",
"MYL9",
# Smooth Muscle
"MYH11",
"CNN1")

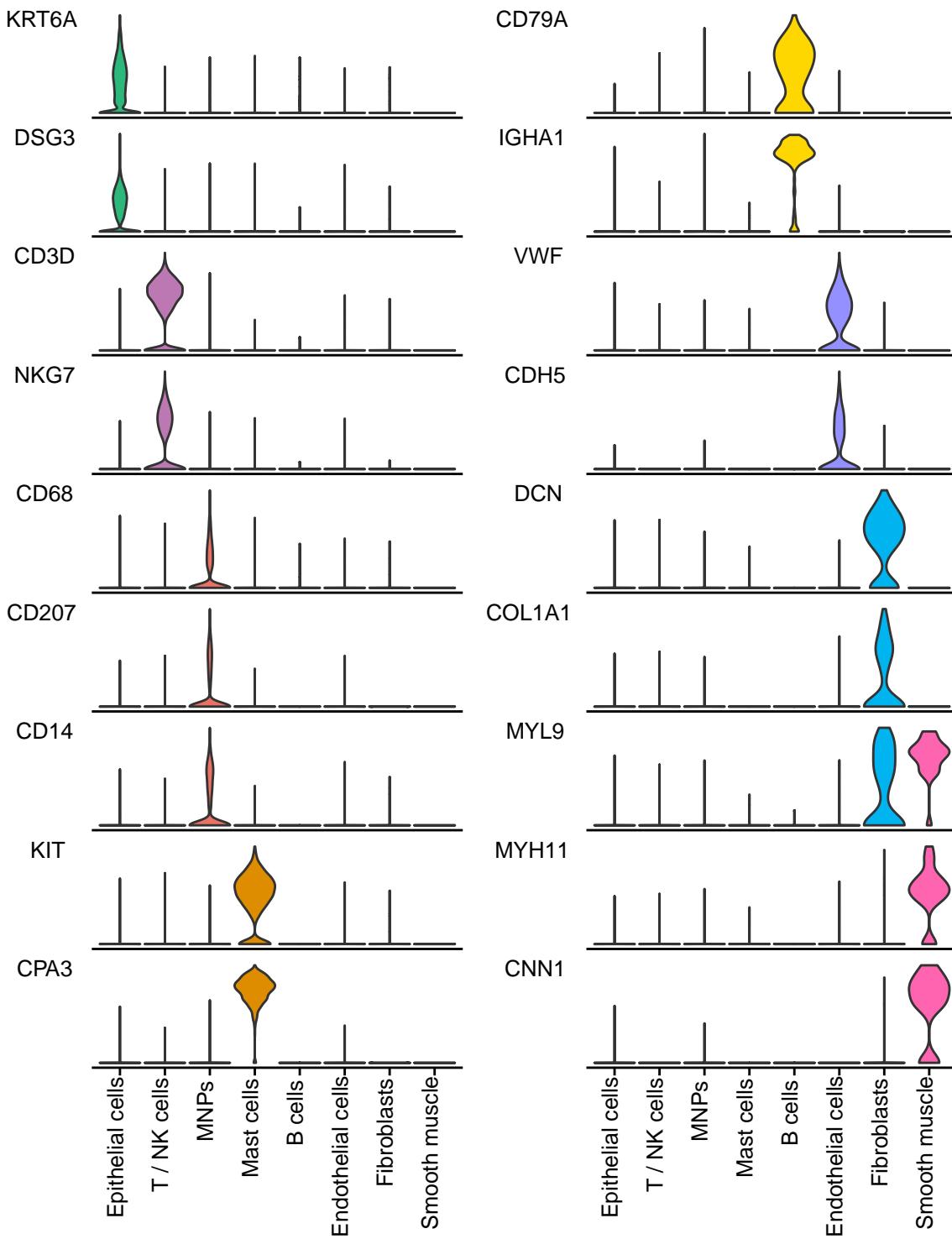
# Assign Coloring
coloring <- c("#2CB77B", "#Epithelial cells"
  "#BC78B2", "#T/NK cells"
  "#E97265", "#MVPs"
  "#DE8C00", "#Mast cells"
  "#FFD700", # B Cells
  "#9590FF", "#Endothelial cells"
  "#00B4FO", "#Fibroblasts"
  "#FF64BO") #Smooth muscle"

# Create Stacked Violin Plot
p1 <- StackedVlnPlot(obj = AllCellTypes_Object,
  features = markers[1:9],
  ident = "CellTypes",
  colors = coloring,
  xlabs = levels(AllCellTypes_Object$CellTypes),
  negNum = 12,
  font = 10)

p2 <- StackedVlnPlot(obj = AllCellTypes_Object,
  features = markers[10:18],
  ident = "CellTypes",
  colors = coloring,
  xlabs = levels(AllCellTypes_Object$CellTypes),
  negNum = 12,
  font = 10)

p1|p2

```



Bar Plot of Cell Type Frequency between HC & EoE

```
# Gather Meta data
MetaD <- AllCellTypes_Object@meta.data %>%
  group_by(DiseaseState) %>%
  sample_n(min(table(AllCellTypes_Object$DiseaseState)), replace = F) %>% # account for differential sample #
  ungroup()

table <- as.data.frame(table(MetaD$CellTypes, MetaD$DiseaseState)) %>%
  rename_all(~c("CellTypes", "DiseaseState", "Frequency")) %>%
  mutate(DiseaseState = factor(DiseaseState, levels = c("EoE_Biopsy", "Healthy_Control")))

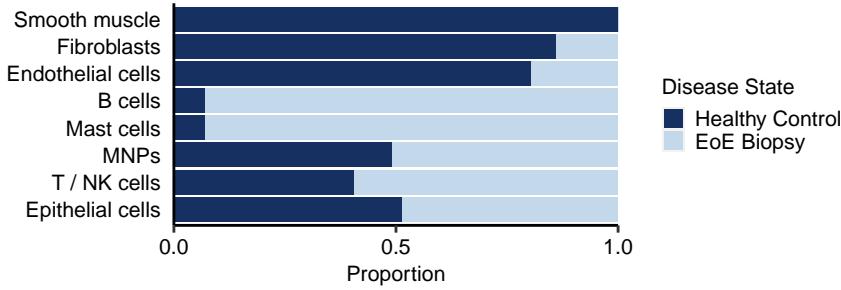
# Construct bar plot
ggplot(table, aes(fill=DiseaseState, y=Frequency, x=CellTypes)) +
  geom_bar(position="fill", stat="identity") +
  coord_flip() +
  scale_fill_manual(values = c("#C3D8EB", "#E64A89", "#163161"), labels = c("EoE Biopsy", "Healthy"))

```

```

        "Healthy Control"),
      name = "Disease State") +
theme(panel.grid.minor = element_blank(),
  panel.background = element_blank(),
  axis.ticks.y = element_blank(),
  axis.line = element_line(colour = "black"),
  axis.text = element_text(size =8,
                           colour = "black"),
  legend.title = element_text(size =8),
  legend.text = element_text(size =8),
  legend.position ="right",
  legend.key.size = unit(0.3, "cm"),
  plot.margin = unit(c(0, 1, 0, 0), "cm"),
  axis.title.x = element_text(size=8),
  aspect.ratio = .5) +
guides(fill = guide_legend(reverse = TRUE)) +
scale_y_continuous(expand = c(0, 0), breaks = c(0,0.5,1)) +
xlab("") + ylab("Proportion")

```



Pie Chart of Cell Type Frequency

```

# Calculate cell type proportions
All_Proportions <- as.data.frame(table(AllCellTypes_Object$CellTypes)) %>%
  mutate(percent = 100*(Freq / ncol(AllCellTypes_Object))) %>%
  arrange(desc(percent)) %>%
  filter(Vari %in% levels(AllCellTypes_Object$CellTypes)) %>%
  mutate(caption = paste(round(percent,2), "% ", Vari, sep = ""))
  mutate(caption = factor(caption, levels = .\$caption))

# Assign coloring
coloring <- c("#2CB77B", "#Epithelial cells",
  "#BC78B2", "#CD8 T cells",
  "#E97265", "#MNPs",
  "#DE8C00", "#Mast cells",
  "#9590FF", "#Endothelial cells",
  "#00B4FO", "#Fibroblasts",
  "#FFD700", # B Cells
  "#FF64B0") #Smooth muscle

# Construct pie chart
ggplot(All_Proportions, aes(x="", y=Freq, fill=caption)) +
  geom_bar(stat="identity", width=1, color="white", size = 0.5) +
  coord_polar("y", start=0) +
  theme_void() +
  theme(legend.title = element_text(size = 8),
    legend.text = element_text(size=8),
    legend.justification = c(.6,0.8),
    aspect.ratio =1,
    legend.key.size = unit(0.5, "cm")) +
  scale_fill_manual(name = "Cell Type",
    values = coloring) +
  ggtitle("")

```

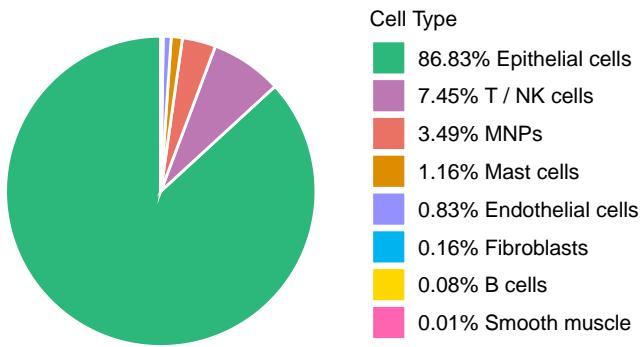
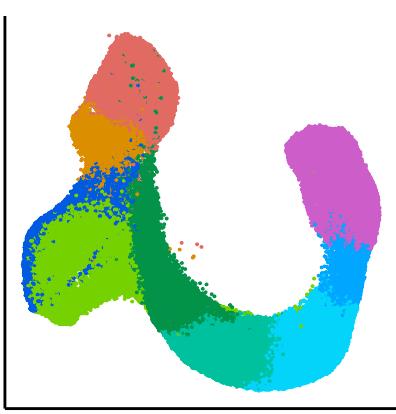


Figure 3

UMAP of Epithelial Clusters

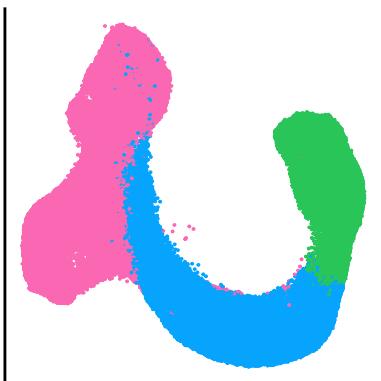
```
# Assign Coloring
coloring <- c("#E16B63", "#Q1
               "#DB8E00", "#Q2
               "#005BEO", "#BD
               "#76D100", "#EB
               "#029349", "#SB1
               "#00C19E", "#SB2
               "#04D3FA", "#SB3
               "#00A6FF", "#SF1
               "#CC5DC9") #SF2

# Construct UMAP plot
left_join(data.frame(Epi_Object@reductions$umap@cell.embeddings) %>% rownames_to_column(var = "Cells"),
          data.frame(Epi_Object@meta.data %>% select(Clusters)) %>% rownames_to_column(var = "Cells")) %>%
  ggplot(aes(x = UMAP_1, y = UMAP_2, color = Clusters)) +
  geom_point(size = 0.01) +
  scale_color_manual(values = coloring,
                     name = "Clusters") +
  theme(axis.line = element_line(colour = "black"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank(),
        plot.margin = unit(c(0, 0, 0, 0), "cm"),
        axis.text.y=element_blank(),
        axis.ticks.x=element_blank(),
        axis.ticks.y=element_blank(),
        axis.text.x=element_blank(),
        aspect.ratio = 1,
        legend.title = element_text(size=7),
        legend.text = element_text(size=7, lineheight = 0.2),
        legend.key=element_rect(size = unit(.1, "lines"),
                               fill = NA)) +
  xlab("") + ylab("") +
  guides(color = guide_legend(override.aes = list(size=3))) +
  ggtitle("")
```



UMAP of Epithelial Compartments

```
# Construct UMAP plot
left_join(data.frame(Epi_Object@reductions$umap@cell.embeddings) %>% rownames_to_column(var = "Cells"),
          data.frame(Epi_Object@meta.data %>% select(Compartments)) %>% rownames_to_column(var = "Cells")) %>%
  ggplot(aes(x = UMAP_1, y = UMAP_2, color = Compartments)) +
  geom_point(size = 0.01) +
  scale_color_manual(values = c("#FA68B4",
                                "#07A4FE",
                                "#2AC559"),
                     name = "Epithelial Compartment") +
  theme(axis.line = element_line(colour = "black"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank(),
        plot.margin = unit(c(0, 0, 0, 0), "cm"),
        axis.text.y=element_blank(),
        axis.ticks.x=element_blank(),
        axis.ticks.y=element_blank(),
        axis.text.x=element_blank(),
        aspect.ratio = 1,
        legend.title = element_text(size=7),
        legend.text = element_text(size=7, lineheight = 0.2),
        legend.key=element_rect(size = unit(.1, "lines"),
                               fill = NA)) +
  xlab("") + ylab("") +
  guides(color = guide_legend	override.aes = list(size=3))) +
  ggtitle("")
```



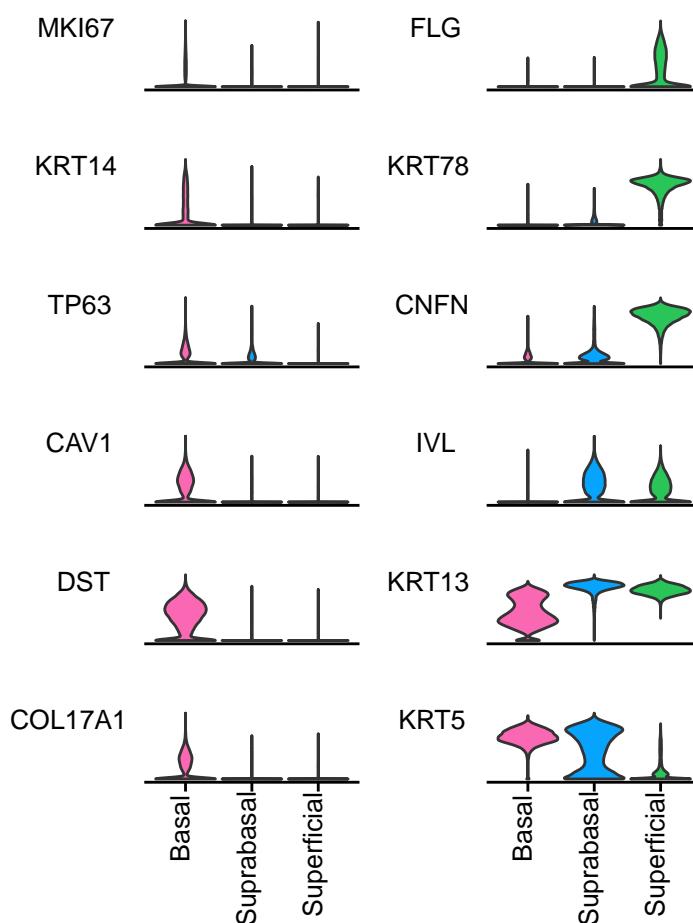
Violin Plot of Epithelial Compartment Markers

```
# Assign Markers
markers <- c("COL17A1", "DST", "CAV1", "TP63", "KRT14", "MKI67", "KRT5", "KRT13", "IVL", "CNFN", "KRT78", "FLG")

### Subset Healthy samples
HC_subset <- Epi_Object[,Epi_Object@meta.data %>% filter(DiseaseState %in% "Healthy_Control") %>% rownames()]

# Generate Stacked Vln Plots
p1 <- StackedVlnPlot(obj = HC_subset,
                      features = rev(markers[c(1:6)]),
                      ident = "Compartments",
                      colors = c("#FA68B4",
                                "#07A4FE",
                                "#2AC559"),
                      xlabs = levels(HC_subset$Compartments),
                      negNum = 12.5,
                      split.by = NULL,
                      ylabs = rev(markers[c(1:6)]),
                      font = 10)
p2 <- StackedVlnPlot(obj = HC_subset,
                      features = rev(markers[7:12]),
                      ident = "Compartments",
                      colors = c("#FA68B4",
                                "#07A4FE",
                                "#2AC559"),
                      xlabs = levels(HC_subset$Compartments),
                      negNum = 12.5,
                      split.by = NULL,
                      ylabs = rev(markers[7:12]),
                      font = 10)

# Display Plots
p1|p2
```



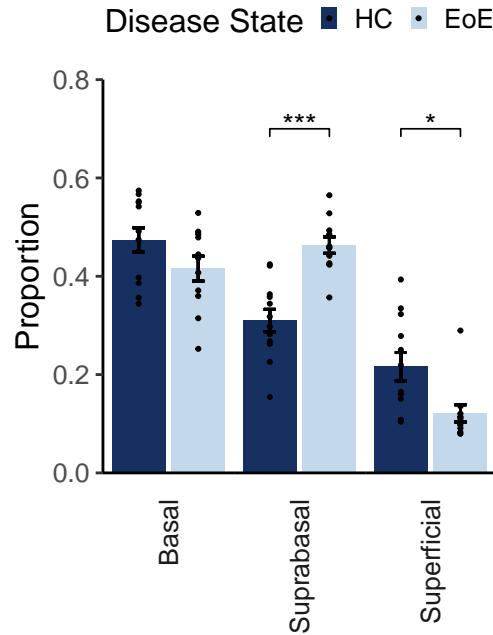
Proportion of Epithelial Compartments

```
# Generate bar plot
Proportion_Barplot_withMultipleComparisons(Object = Epi_Object,
                                             ClusterIdent = "Compartments",
                                             Cat_List = list("Healthy_Control",
                                                             "EoE_Biopsy"),
                                             Category_Ident = "DiseaseState",
                                             my_comparisons = list(c("Healthy_Control", "EoE_Biopsy"))),
```

```

        colors = c("#163161", "#HC
                   "#C3D8EB"),
        remove_position = NULL,
        GroupLabels = as.vector(levels(Epi_Object$Compartments)),
        yhi = 0.8,
        ypos = 0.7,
        stepval =0,
        xlab_angle = 90,
        font =10,
        NewNames = c("HC","EoE"),
        hjust = 1,
        bracket_length =.025) +
theme(axis.ticks.x = element_blank(),
      aspect.ratio = 3,
      legend.position = "top",
      legend.justification = "center",
      legend.key.size = unit(0.25, "cm"))

```



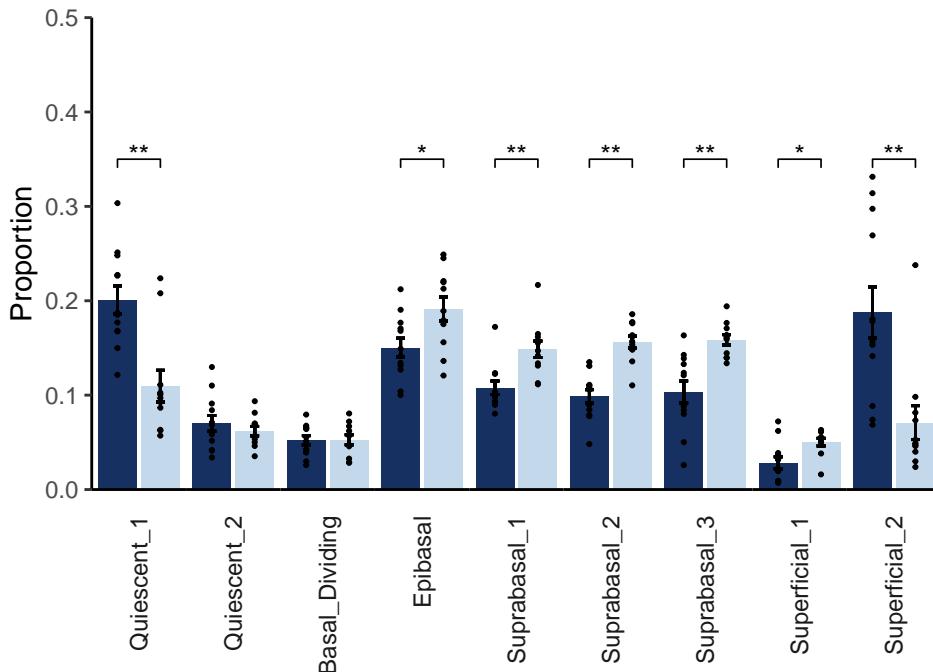
Proportion of Epithelial Clusters

```

# Generate bar plot
Proportion_Barplot_withMultipleComparisons(Object = Epi_Object,
                                             ClusterIdent = "Clusters",
                                             Cat_List = list("Healthy_Control",
                                                             "EoE_Biopsy"),
                                             Category_Ident = "DiseaseState",
                                             my_comparisons = list(c("Healthy_Control","EoE_Biopsy")),
                                             colors = c("#163161", #HC
                                                       "#C3D8EB"),
                                             remove_position = NULL,
                                             GroupLabels = as.vector(levels(Epi_Object$Clusters)),
                                             yhi = 0.5,
                                             ypos = 0.35,
                                             stepval =0,
                                             xlab_angle = 90,
                                             font =9,
                                             NewNames = c("HC","EoE"),
                                             hjust = 1,
                                             bracket_length =.025) +
theme(axis.ticks.x = element_blank(),
      aspect.ratio = 5,
      legend.position = "top",
      legend.justification = "center",
      legend.key.size = unit(0.25, "cm"))

```

Disease State ■ HC ● EoE



MKI67 Violin Plot

```
# Generate Vln Plot
VlnPlot(Epi_Object,
  group.by = "Clusters",
  split.by = "DiseaseState",
  pt.size = 0,
  features = "MKI67") +
  ggtitle("") +
  xlab("") +
  ylab("MKI67") +
  theme(legend.position="right",
    plot.title = element_text(size=20),
    axis.text.x = element_text(angle=90,
      hjust=1,
      size = 10,
      vjust = 0.5),
    axis.text.y = element_text(size=10),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line.x.bottom = element_line(color="black"),
    axis.line.y.left = element_line(color="black"),
    panel.border = element_blank(),
    strip.background = element_blank(),
    strip.placement = "outside",
    strip.text = element_text(size=10),
    legend.text=element_text(size=10),
    legend.title=element_text(size=10),
    axis.title=element_text(size=10),
    panel.spacing = unit(0, "lines"),
    aspect.ratio = .3) +
  scale_fill_manual(values = c("#163161",
    "#C3D8EB"),
    labels = c("HC", "EoE"))
```

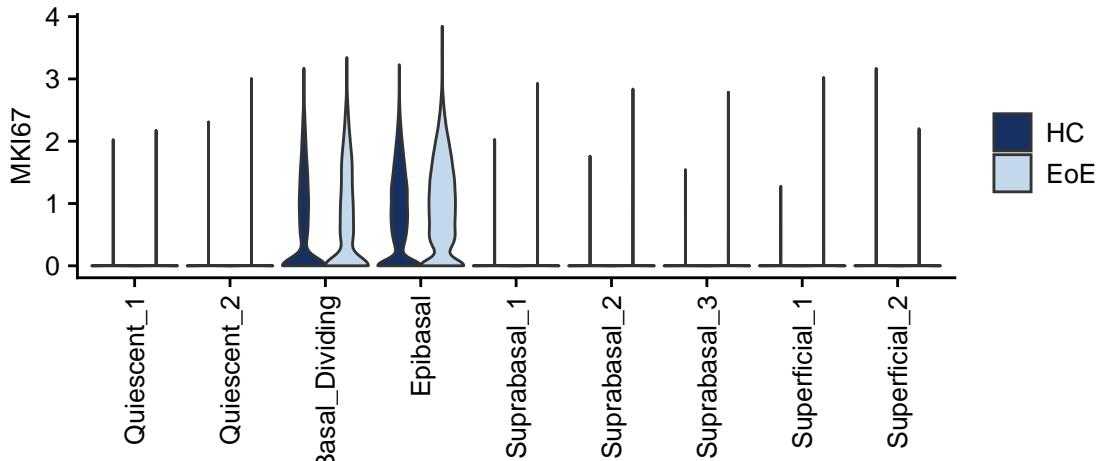


Figure 4

DotPlot of Epithelial Clusters

```
# Assign markers
markers <- c(
  "SOX2", "TP63", "KLF5", # Progenitor / Basal
  "KRT13", "IVL", #Early Differentiation / Suprabasal
  "CNFN", "KRT78", "FLG" # Terminal Differentiation / Superficial
)

# Generate base plot
p1 <- DotPlot(Epi_Object,
  group_by = "Clusters_byDiseaseState",
  features = markers ) +
  coord_flip(clip="off") +
  theme(axis.text.x = element_text(angle=90, hjust=1, vjust = 0.5),
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    plot.margin = unit(c(2,0,2,0), "lines"),
    aspect.ratio = 0.7,
    legend.position = "top",
    legend.justification = "center") +
  xlab("") +
  ylab("") +
  scale_y_discrete(labels = rep(levels(Epi_Object$Clusters),2)) +
  scale_colour_gradient2(low="#382C84", mid="lightgrey", high="#EF0053",
    breaks = c(-1.5,0,1.5))

# Annotate plot

## Generate grobs for disease state labels
text1 <- grid::textGrob("HC",
  x = (0.25),
  y =(.56),
  hjust = 0.5,
  gp = gpar(cex = 1,
            fontface="bold"))
line1 <- grid::linesGrob(unit(c(0.0, 0.49), "npc"),
  unit(c(.57, .57), "npc"),
  gp = gpar(lwd = 2))

text2 <- grid::textGrob("EoE",
  x = (0.75),
  y =(.56),
  hjust = 0.5,
  gp = gpar(cex = 1,
            fontface="bold"))
line2 <- grid::linesGrob(unit(c(0.51, 1), "npc"),
  unit(c(.57, .57), "npc"),
  gp = gpar(lwd = 2))
```

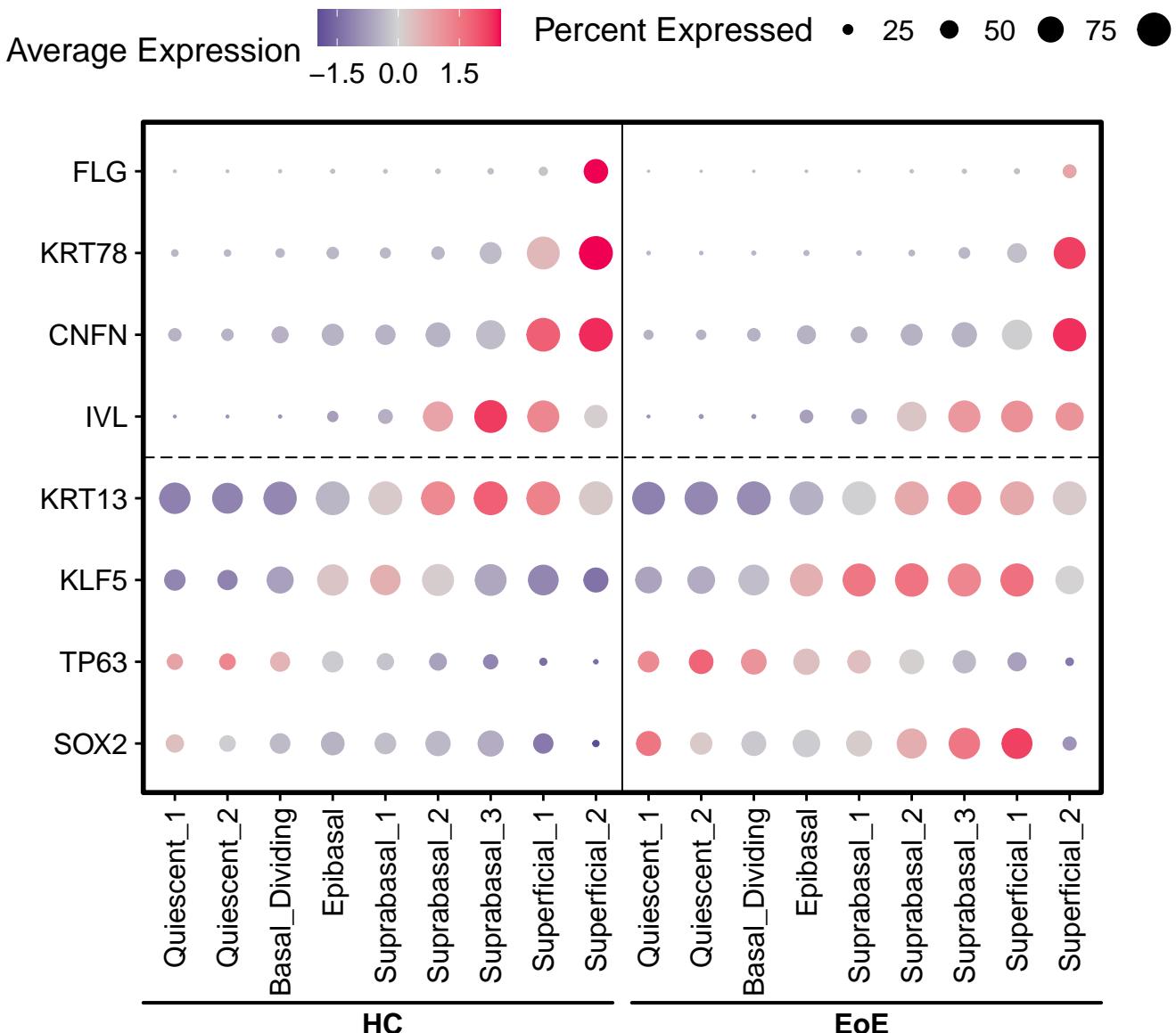
```

## Attach annotation to plot
p2 <- p1 + annotation_custom(combo,
                               xmin = -16.45,
                               xmax = Inf,
                               ymin= -Inf,
                               ymax= Inf)

## Divide plot area for readability
gline <- linesGrob(x=0:1,
                     y=.5,
                     gp=gpar(cex=4,
                             lty='longdash'))
gline_vert <- linesGrob(y=0:1,
                        x=.5,
                        gp=gpar(cex=4))
p3 <- p2 + annotation_custom(gline,
                               xmin = -Inf,
                               xmax = Inf,
                               ymin = -Inf,
                               ymax = Inf) +
  annotation_custom(gline_vert,
                    xmin = -Inf,
                    xmax = Inf,
                    ymin = -Inf,
                    ymax = Inf)

# Display annotated plot
p3

```



UpSet Plot for Epithelial Cluster DEGs

```
# Load and filter DEG lists per epithelial cluster
Clusters <- levels(Epi_Object$Clusters)
deg_lists <- list()
gene_list <- data.frame()
setwd("/projects/b1042/MPLab/mhc0155/EoE_Project")

for(x in 1:length(Clusters)){
  # Load in all DEG results per cluster
  list <- loadRData(paste("DEG/Clusters/",
    Clusters[x],
    "_DEG_",
    "EoE_toHC",
    sep = ""))
  
  # Df of DEGs per epithelial cluster, with filtering
  listF <- list %>%
    filter(abs(Log2FC) > 0.25 & FDR_Pval < 0.05) %>%
    filter(pct.EoE_Biopsy > 15 | pct.Healthy_Control > 15)
  deg_lists[[x]] <- listF$gene
  
  # DF of distinct DE genes across clusters, post-filter
  listF2 <- listF %>%
    mutate(Cluster = Clusters[x])
  if (x == 1) {
    gene_list <- listF2 %>% distinct(gene, .keep_all = TRUE)
  } else {
    gene_list <- rbind(gene_list, listF2) %>% distinct(gene, .keep_all = TRUE)
  }
}

# Create binary matrix of gene presence in each set - all genes

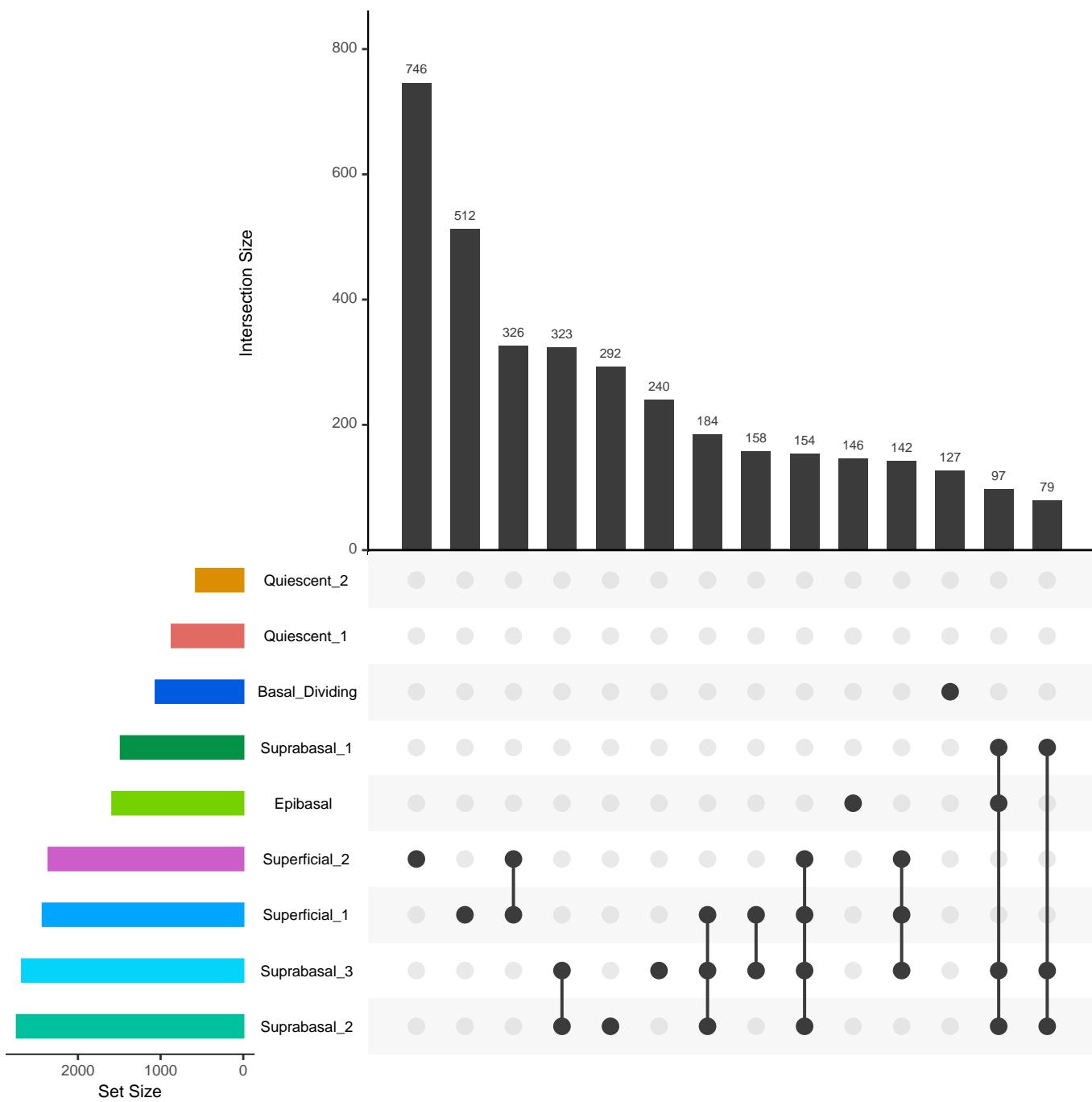
deg_matrix <- matrix(0, nrow = length(gene_list$gene), ncol = length(deg_lists)) #rows = genes, cols = clusters
for(i in 1:length(deg_lists)) {
  deg_matrix[match(deg_lists[[i]], gene_list$gene), i] <- 1
}

colnames(deg_matrix) <- Clusters
rownames(deg_matrix) <- gene_list$gene
deg_matrix <- as.data.frame(deg_matrix)

# Generate UpSet plot

coloring <- c("#DB8E00",
  "#E16B63",
  "#005BEO",
  "#029349",
  "#76D100",
  "#CC5DC9",
  "#00A6FF",
  "#04D3FA",
  "#00C19E")

upset(deg_matrix,
  nssets = 9,
  nintersects = 14,
  mb.ratio = c(0.5, 0.5),
  order.by = c("freq"),
  decreasing = c(TRUE, FALSE),
  sets.bar.color = rev(coloring),
  point.size = 3.5)
```



Heatmap of epithelial DEGs

```
# Load in DEGs across all epithelial cells, calculated as EoE vs HC
setwd("/projects/b1042/MPLab/mhc0155/EoE_Project")
DEG_AllEpis_EoEtoHC <- loadRData("DEG/AllEpis/AllEpisDEGs_EoEtoHC")

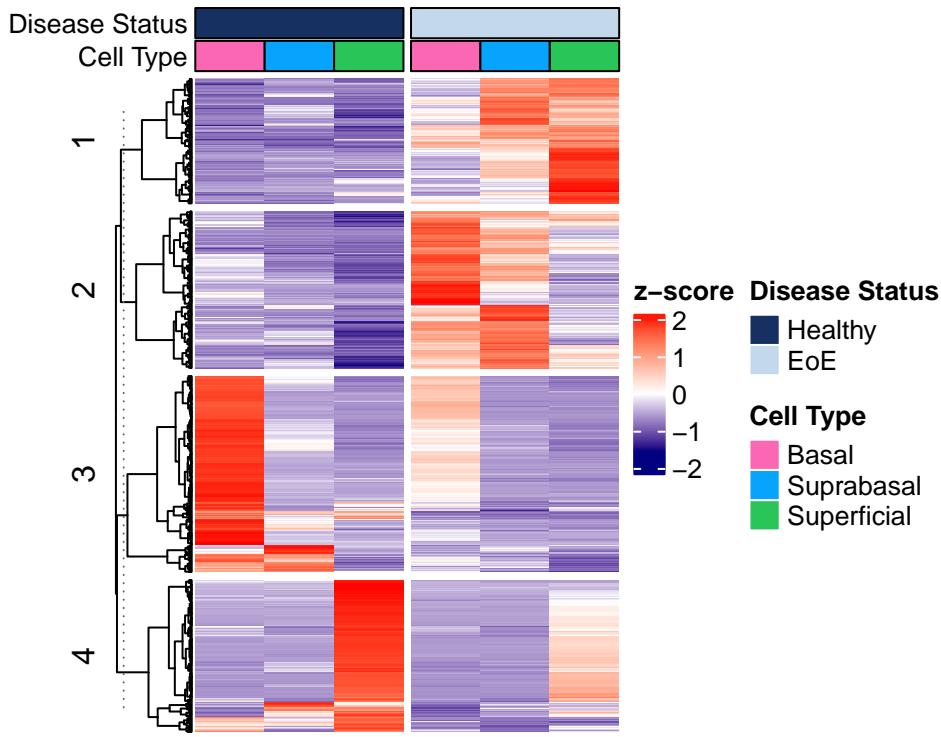
# Filter DEGs
DEG_AllEpis_EoEtoHC_filtered <- DEG_AllEpis_EoEtoHC %>%
  filter(FDR_Pval < 0.05,
         abs(Log2FC) > 1)

# Extract normalized RNA assay
DEG_exp <- AverageExpression(Epi_Object,
                               assay = "RNA",
                               slot = "data",
                               group_by = "Compartments_byDiseaseState",
                               features = sort(rownames(DEG_AllEpis_EoEtoHC_filtered)))

# Function to scale a df colwise
ScaleColumns <- function(input_df){

  # Stash new df to store scaled values
  input_df_scaled <- input_df

  # Scale each column of values
  columncount <- ncol(input_df)
  for(i in 1:columncount){
    columnnum <- colnames(input_df)[i]
    input_df[,columnnum] <- as.numeric(input_df[,columnnum])
  }
}
```

```
# Extract genes from each cluster
r.dend <- row_dend(HM) #Extract row dendrogram
rcl.list <- row_order(HM) #Extract clusters (output is a list)

## Extract gene names from each cluster as a list of vectors
Clusters <- paste0("Cluster ", names(rcl.list))
geneLIST <- list()
for (i in 1:length(rcl.list)){
  clusterNum <- names(rcl.list)[i]
  if (i == 1) {
    clu <- t(t(row.names(DEG_exp_scaled[rcl.list[[i]],])))
    geneLIST[[i]] <- as.vector(clu)
  } else {
    clu <- t(t(row.names(DEG_exp_scaled[rcl.list[[i]],])))
    geneLIST[[i]] <- as.vector(clu)
  }
}

## Convert cluster list to matrix
NewMaxP <- max(sapply(geneLIST, length))
AddedSpacesP <- lapply(geneLIST, function(X) {
  c(as.character(X), rep("", times = NewMaxP - length(X)))
})
Genes_perCluster <- as.data.frame(do.call(cbind, AddedSpacesP))
colnames(Genes_perCluster) <- Clusters

## Gather DE info for the genes in each cluster. Used as input to metascape for pathway analysis.
Cluster_Expression <- Genes_perCluster %>% pivot_longer(cols = Clusters,
  names_to = "Cluster",
  values_to = "gene") %>%
  merge(x = ., y = DEG_AllEpis_EoEtoHC_filtered, by = "gene") %>%
  arrange(Cluster, desc(Log2FC))

setwd("/projects/b1042/MPLab/mhc0155/EoE_Project/DEG/AllEpis/HM")
save(Cluster_Expression, file = "AllEpisDEGs_EoEtoHC_HMacrossCompartments_GenesperCluster")

# Display genes per cluster
kable(head(Genes_perCluster, 15))
```

Cluster 1	Cluster 2	Cluster 3	Cluster 4
ALOX15	TNFAIP6	CXCL12	DEFB4A
LRRC31	NTRK1	LUARIS	GABRB2
PMCH	POSTN	HOXD9	FAM25G
CCL26	NRXN1	NNMT	KCTD4
BANCR	ADAMTS15	LPL	SPRR2B
SHD	ATP13A5	ZNF98	SPRR2F
IL13	ST18	CDH12	CD177
AFF2	QRFPR	CLSTN2	HTR3A
DPP4	ISL1	NAV3	FAM25C
CMA1	IL13RA2	ABCA8	SOX17

Figure 5

EnrichR analysis of Cluster 1 DEGs

```
# Load DEGs from Cluster 1 for analysis (from Fig 4C)
setwd("/projects/b1042/MPLab/mhc0155/EoE_Project/DEG/AllEpis/HM/")

AllEpi_DEGs <- loadRData("AllEpisDEGs_EoEtoHC_HMacrossCompartments_GenesperCluster")
Clus1_DEGs <- (AllEpi_DEGs %>% filter(Cluster %in% "Cluster 1"))$gene

# Run EnrichR analysis

## Set human genes
setEnrichrSite("Enrichr") # Human genes

## Access database options & set databases
dbs <- listEnrichrDbs()
EnrichR_lib <- c("ChEA_2022", "TF_Perturbations_Followed_by_Expression")

## Run enrichment and plot top hits
for(h in 1:length(EnrichR_lib)){

  # Calculate Enrichment
  EnrichR_Test <- enrichr(clus1_DEGs, EnrichR_lib[h])
  save(EnrichR_Test,
    file = gsub(" ", "",
    "",
    paste("EnrichR_Test_Cluster1_",
      EnrichR_lib[h],
      sep = "")))

  # Curate df in terms of top predicted, ranked by adjusted p value
  NonExpressedGenes <- c("FOXH1", "GATA1", "GATA2", "HNF1A", "MNX1", "NANOG", "LXR", "RNF2", "RUNX1", "RUNX", "SALL4", "SRY",
    "SCL", "SMRT", "FOXA2", "FOXP3", "GATA3", "GFI1B", "HMGA2", "IRF5", "NR5A2", "PPARG", "TFAP2C", "TAL1")

  EnrichrDF <- EnrichR_Test[[1]] %>%
    mutate(gene = sapply(Term, function(x) strsplit(x, " ")[[1]][1])) %>%
    arrange(Adjusted.P.value) %>%
    distinct(gene, .keep_all = T) %>%
    filter(gene %in% NonExpressedGenes) %>%
    dplyr::slice(1:10) %>%
    tidyverse::separate(Overlap, c("numgenes", "numgenes_inpathway"), sep = "/") %>%
    mutate(percent_genes = 100*as.numeric(numgenes)/as.numeric(numgenes_inpathway),
      neglog10adjP = as.numeric(-log10(as.numeric(Adjusted.P.value))),
      gene = as.character(gene),
      gene = factor(gene, levels=rev(gene)),
      numgenes = as.numeric(numgenes))

  save(EnrichrDF,
    file = gsub(" ", "",
    "",
    paste("EnrichR_Test_Cluster1_",
      EnrichR_lib[h],
      "_Filtered",
      sep = "")))

  # Plot top predicted results
  low = min(EnrichrDF$neglog10adjP)
  middle = mean(EnrichrDF$neglog10adjP)
  high = max(EnrichrDF$neglog10adjP)
  buffer = (high-low)*0.15

  plot <- EnrichrDF %>%
    ggplot(aes(x=gene, y=numgenes, fill=neglog10adjP)) +
    geom_bar(stat="identity") +
    coord_flip() +
    theme_bw() +
    theme(panel.grid.major.x = element_blank(),
      panel.grid.minor.x = element_blank(),
      panel.grid.major.y = element_blank(),
      axis.text.x = element_text(size = 8),
      axis.text.y = element_text(size = 8),
      legend.title = element_text(size=8),
      legend.text = element_text(size=8),
      panel.border = element_blank(),
      axis.title.x = element_text(size=8),
      plot.title = element_text(size=8),
      axis.line.x.bottom = element_line(color = "black"),
      axis.line.y.left = element_line(color = "black"),
      aspect.ratio = .7) +
    xlab("") +
    ylab("Gene count") +
    scale_fill_gradientn(colours = rev(c("#FD0010", "#B100AA", "#0000FF")),
      values = rescale(c(low, middle, high)),
      guide = "colorbar", limits=c(low,high),
      name = "-log10(Padj)" ,
      breaks = unlist(map(seq(low + buffer, high - buffer, by=(high-low - 2*buffer)/2), .f = function(x){round(x, 2)}))) +
    ggtitle(paste("Cluster 1",
      " ",
      EnrichR_lib[h]))

  # Display plot
  print(plot)
}
```

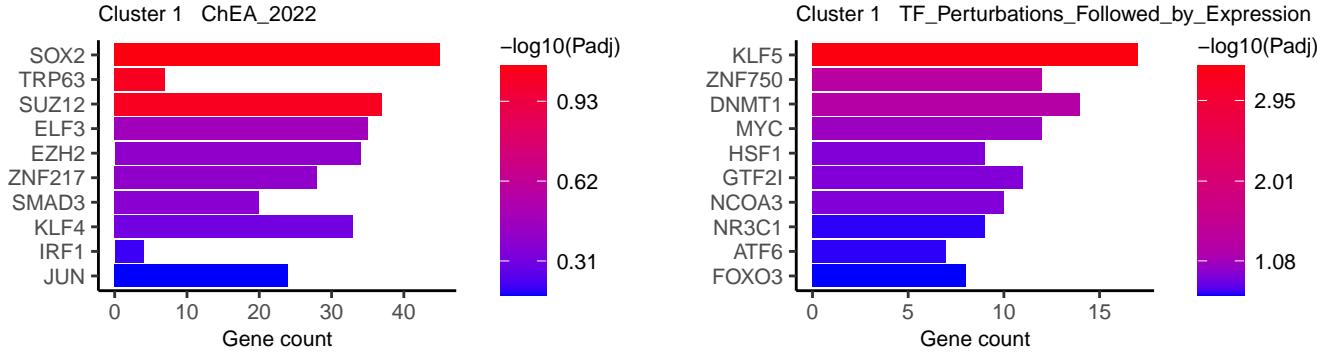
```

# Save plot

plot

pdf(file = gsub(" ", 
  "", 
  paste("EnrichR_Test_Cluster1_",
    EnrichR_lib[h],
    "_TopResults.pdf",
    sep = "")),
  width = 10,
  height = 10)
dev.off()
}

```



Violin plots of TFs and downstream targets

```

# Load EnrichR dfs
setwd("/projects/b1042/MPLab/mhc0155/EoE_Project/DEG/AllEpis/HM")
ChEA_df <- loadRData("EnrichR_Test_Cluster1_ChEA_2022_Filtered")
TFPeturb_df <- loadRData("EnrichR_Test_Cluster1_TF_Perturbations_Followed_by_Expression_Filtered")

# Extract genes
TFs <- c("SOX2",
  "KLF5",
  "TP63",
  "KLF4",
  "SUZ12")
TF_Targets <- list("SOX2" = unlist(strsplit((ChEA_df %>% dplyr::filter(gene %in% "SOX2"))$Genes, ";")),
  "KLF5" = unlist(strsplit((TFPeturb_df %>% dplyr::filter(gene %in% "KLF5"))$Genes, ";")),
  "TP63" = unlist(strsplit((ChEA_df %>% dplyr::filter(gene %in% "TP63"))$Genes, ";")),
  "KLF4" = unlist(strsplit((ChEA_df %>% dplyr::filter(gene %in% "KLF4"))$Genes, ";")),
  "SUZ12" = unlist(strsplit((ChEA_df %>% dplyr::filter(gene %in% "SUZ12"))$Genes, ";")))

# Add gene signatures for DEG targets
for(x in 1:length(TFs)){
  Epi_Object <- AddModuleScore(object = Epi_Object,
    features = list("TF" = TF_Targets[[TFs[x]]]),
    name = paste0(TFs[x], "_Targets_geneSignature"))
}

# Plot compartment TF expression and target gene signature scores
p1 <- StackedVlnPlot(obj = Epi_Object,
  features = TFs,
  ident = "Compartments",
  colors = c("#163161", "#HC
  "#C3D8EB"),
  xlabs = levels(Epi_Object$Compartments),
  split.by = "DiseaseState",
  negNum = 12.5,
  ylabs = TFs,
  font = 10)

p2 <- StackedVlnPlot(obj = Epi_Object,
  features = paste0(TFs, "_Targets_geneSignature1"),
  ident = "Compartments",
  colors = c("#163161", "#HC
  "#C3D8EB"),
  negNum = 12.5,
  ylabs = TFs,
  font = 10)

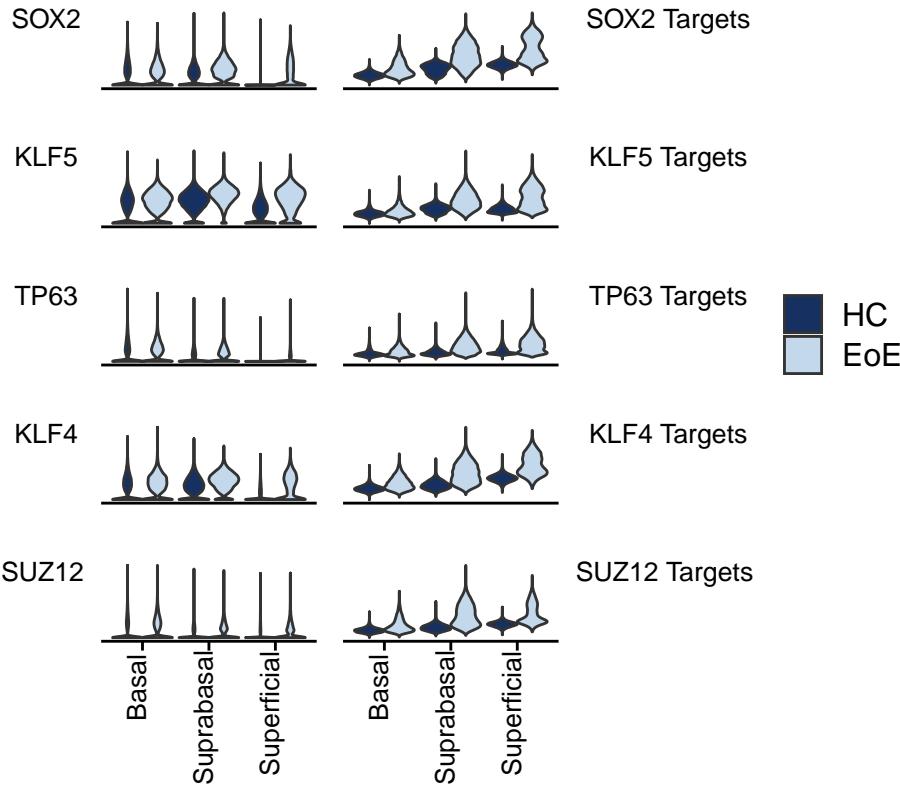
```

```

xlab = levels(Epi_Object$Compartments),
negNum = 12.5,
split.by = "DiseaseState",
ylab = paste0(TFs, " Targets"),
font = 10,
title = "right",
legend = "right",
legendLabs = c("HC", "EoE"))

```

p1 | p2



```
# Extract Disease State specific gene signature data
```

```

Data <- Epi_Object@meta.data %>%
  select(DiseaseState, PatientID, Compartments, Clusters, Quiescent_Sig, Superficial_Sig) %>%
  mutate(Clusters = ifelse(Clusters %in% as.vector(levels(Epi_Object$Clusters))[1:4], "Basal", as.character(Clusters)))

HC_Data <- Data %>%
  filter(DiseaseState %in% "Healthy_Control")

EoE_Data <- Data %>%
  filter(DiseaseState %in% "EoE_Biopsy")

# Set variables for contour plots
textsize = 8.5
leg = c(0.7, 0.63)
binA = 12
binB = 6

CompartmentColors = c("#FA68B4",
                      "#07A4FE",
                      "#2AC559")

ClusterColors <- c("#FA68B4",
                   "#00FF00",
                   "#00CCFF",
                   "#003399",
                   "#CC00FF",
                   "#FF0000")

Contour_theme <- theme(panel.background = element_blank(),
                       panel.grid.major = element_line(),
                       panel.grid.minor = element_line(),
                       aspect.ratio = 1,
                       axis.line.y.left = element_blank(),
                       axis.line.y.right = element_blank(),
                       axis.line.x.top = element_blank(),
                       axis.line.x.bottom = element_blank(),
                       panel.border = element_rect(color = "black", fill = NA, size = 1.5),
                       plot.title = element_text(hjust = 0.5),

```

```

        legend.position = leg,
        legend.key = element_rect(colour = "transparent",
                                   fill = "white",
                                   size = 0.25),
        legend.title = element_text(size=textsize),
        legend.text = element_text(size=textsize-1.5),
        axis.text = element_text(size = textsize),
        axis.title = element_text(size = textsize))

# Construct contour plots, seperated by compartments

p1 <- ggplot(HC_Data, aes(y=Quiescent_Sig, x=Superficial_Sig)) +
  geom_density_2d(bins =binA,aes(colour = Compartments)) +
  Contour_theme +
  scale_color_manual(name = "Compartments",
                     values = CompartmentColors) +
  xlab("")+
  ylab("Quiescent Signature Score")+
  guides(colour = guide_legend(override.aes = list(size=2))) +
  ggtitle("HC")

p2 <- ggplot(EoE_Data, aes(y=Quiescent_Sig, x=Superficial_Sig)) +
  geom_density_2d(bins =binA,aes(colour = Compartments)) +
  Contour_theme +
  scale_color_manual(name = "Compartments",
                     values = CompartmentColors) +
  xlab("Superficial Signature Score")+
  ylab("Quiescent Signature Score")+
  guides(colour = guide_legend(override.aes = list(size=2))) +
  ggtitle("EoE")

# Construct contour plots, seperated by clusters

p3 <- ggplot(HC_Data, aes(y=Quiescent_Sig, x=Superficial_Sig)) +
  geom_density_2d(bins =binB,aes(colour = Clusters)) +
  Contour_theme +
  scale_color_manual(name = "Clusters",
                     values = ClusterColors) +
  xlab("")+
  ylab("")+
  # guides(colour = guide_legend(override.aes = list(size=2))) +
  ggtitle("HC")

p4 <- ggplot(EoE_Data, aes(y=Quiescent_Sig, x=Superficial_Sig)) +
  geom_density_2d(bins =binB,aes(colour = Clusters)) +
  Contour_theme +
  scale_color_manual(name = "Clusters",
                     values = ClusterColors) +
  xlab("Superficial Signature Score")+
  ylab("")+
  # guides(colour = guide_legend(override.aes = list(size=2))) +
  ggtitle("EoE")

# Display Plots

(p1/p2) | (p3/p4)

```

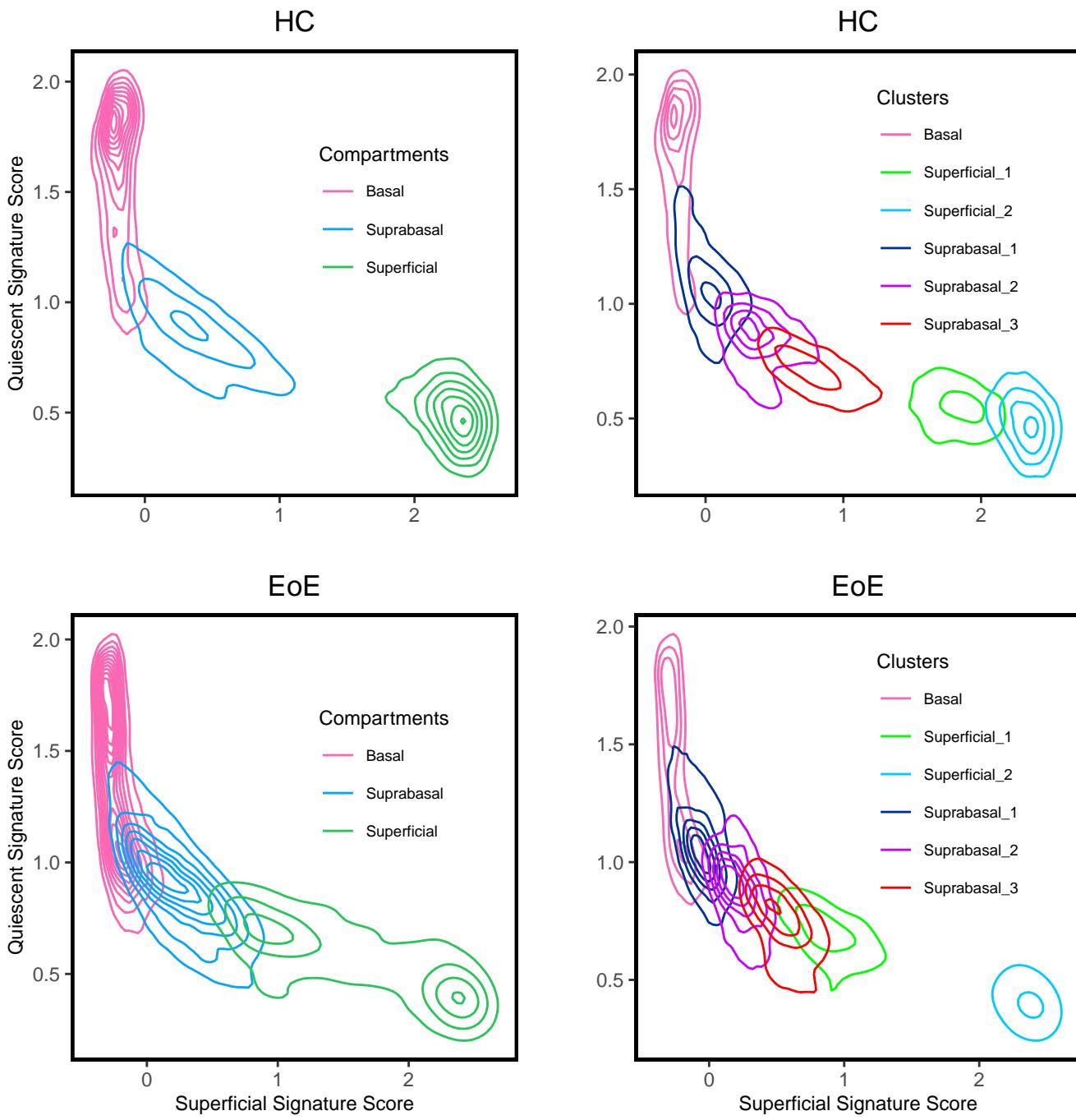


Figure 6

Violin plots of epithelial markers

```
### Construct stacked violin plot
```

```
Markers <- c("CNFN",
           "IVL",
           "TP63",
           "KRT14")
```

```
StackedVlnPlot(obj = Epi_Object,
                features = Markers,
                ident = "Clusters",
                colors = c("#163161", #HC
                          "#C3D8EB"),
                xlabs = levels(Epi_Object$Clusters),
                split.by = "DiseaseState",
                negNum = 12.5,
                ylabs = Markers,
                font = 10,
                legend = "right",
                legendLabs = c("HC", "EoE"))
```

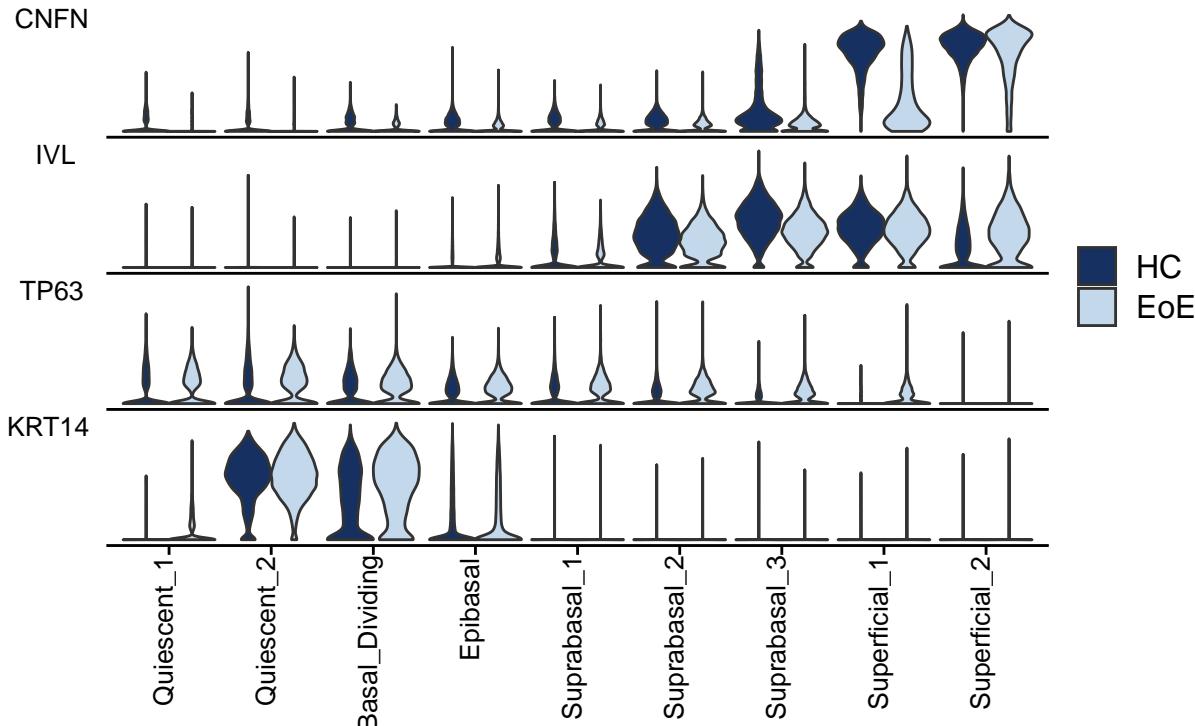
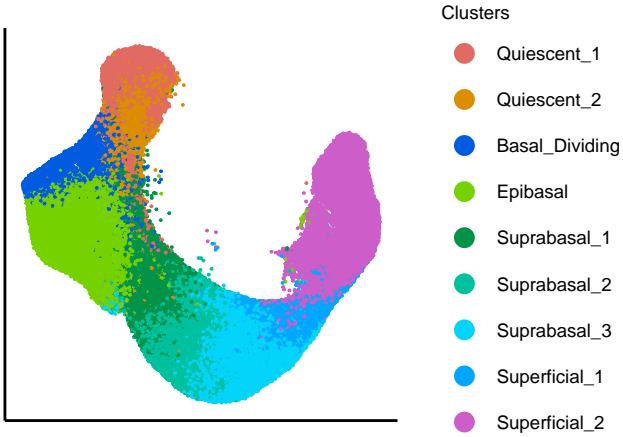


Figure 7

Merged UMAP

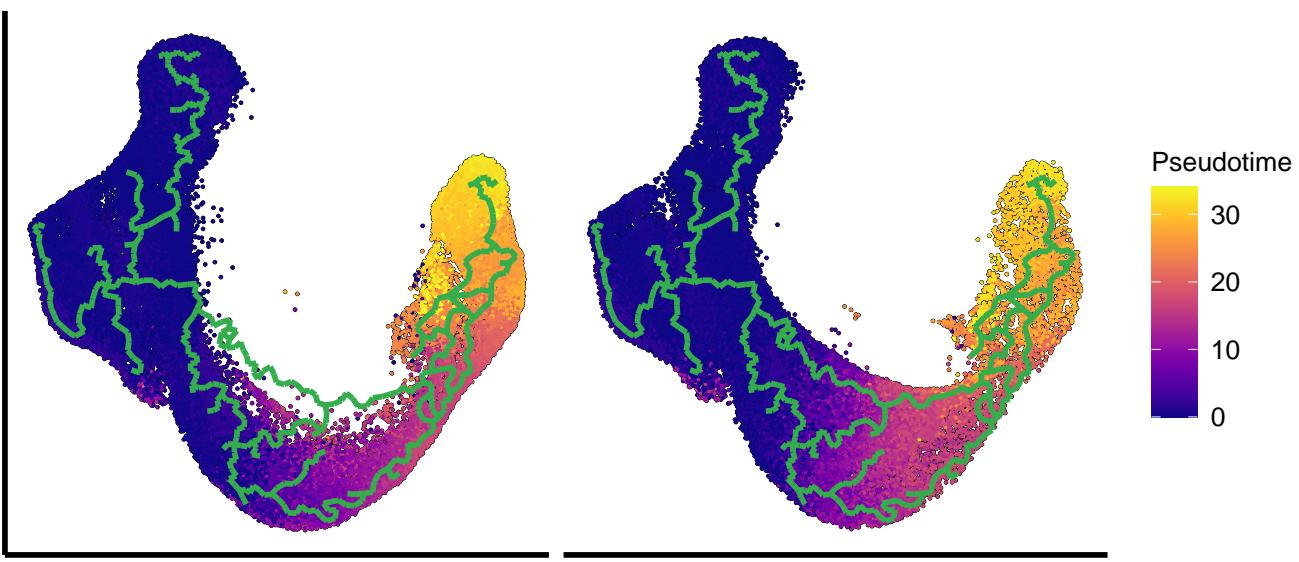
```
# Assign Coloring
coloring <- c("#E16B63", "#B1
  "#DB8E00", "#B2
  "#00B8E0", "#B3
  "#76D100", "#B4
  "#029349", "#B5
  "#00C19E", "#B6
  "#04D3FA", "#B7
  "#00A6FF", "#B8
  "#CC5DC9", "#B9
  "#FF63B6") #B10

# Construct DimPlot
left_join(data.frame(Epi_Object_Merged@reductions$umap@cell.embeddings) %>% rownames_to_column(var = "Cells"),
  data.frame(Epi_Object_Merged@meta.data %>% select(Clusters)) %>% rownames_to_column(var = "Cells")) %>%
ggplot(aes(x = UMAP_1, y = UMAP_2, color = Clusters)) +
  geom_point(size = 0.01) +
  scale_color_manual(values = coloring,
    name = "Clusters") +
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    plot.margin = unit(c(0, 0, 0, 0), "cm"),
    axis.text.y=element_blank(),
    axis.ticks.x=element_blank(),
    axis.ticks.y=element_blank(),
    axis.text.x=element_blank(),
    aspect.ratio = 1,
    legend.title = element_text(size=7),
    legend.text = element_text(size=7, lineheight = 0.2),
    legend.key=element_rect(size = unit(.1, "lines"),
      fill = NA)) +
  xlab("") + ylab("") +
  guides(color = guide_legend(override.aes = list(size=3))) +
  ggtitle("")
```



UMAP with Pseudotime trajectory & values

```
# Construct UMAP with overlaid trajectory
plot_cells(Epi_CDS,
  group_label_size = 0,
  color_cells_by="Pseudotime",
  label_cell_groups = F,
  label_roots = F,
  label_branch_points = F,
  label_leaves = F,
  trajectory_graph_color = "#34ad4d",
  trajectory_graph_segment_size = 1,
  rasterize =F) +
facet_wrap(~DiseaseState, labeller =as_labeller(c(Healthy_Control = "HC",
  EoE_Biopsy = "EoE")))+ 
theme(axis.line = element_line(colour = "black"),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  panel.border = element_blank(),
  panel.background = element_blank(),
  plot.margin = unit(c(0, 0, 0, 0), "cm"),
  axis.text.y=element_blank(),
  axis.ticks.x=element_blank(),
  axis.ticks.y=element_blank(),
  axis.text.x=element_blank(),
  aspect.ratio = 1,
  plot.title = element_text(size=14),
  legend.title = element_text(size=10),
  legend.text = element_text(size=10),
  legend.key=element_blank(),
  axis.line.x.bottom = element_line(color = "black", size =1),
  axis.line.y.left = element_line(color = "black", size =1),
  strip.text = element_text(size=14, face = "bold"))+
scale_x_reverse() +
xlab("") + ylab("")
```



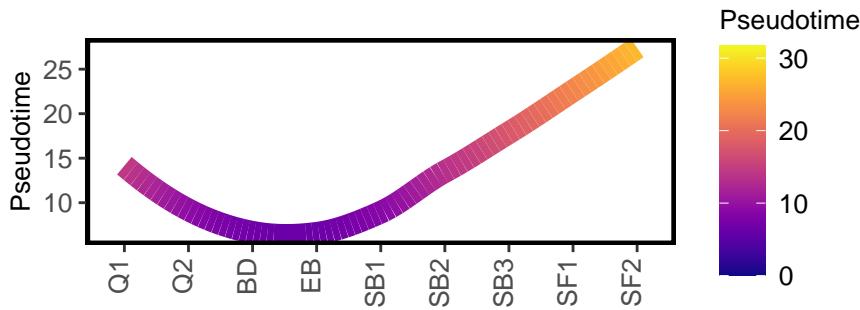
Average pseudotime per epithelial cluster at baseline conditions

```
# Construct df with pseudotime info
df_Pseudo <- FetchData(Epi_Object,
  vars = c("Clusters",
    "Pseudotime",
    "DiseaseState")) %>%
filter(DiseaseState == "Healthy_Control") %>%
group_by(Clusters) %>%
summarise(PseudoAvg = mean(Pseudotime)) %>%
ungroup()

# Construct viridis coloring scale that is reflective of all cells, not just pseudotime cluster averages
color <- viridis(option= "C",
  n = round(max(max(Epi_Object@meta.data$Pseudotime),0)))

# Plot cluster pseudotime distribution
p1 <- df_Pseudo %>%
ggplot(aes(x=Clusters, y=PseudoAvg, group=1)) +
  geom_smooth(aes(color=..y..),
    se=FALSE,
    method = "loess",
    size = 4,
    span = 1) +
  scale_color_gradientn(name = "Pseudotime",
    colors = color,
    limits = c(0,max(Epi_Object@meta.data$Pseudotime))) +
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.background = element_blank(),
    plot.margin = unit(c(0, 0, 0, 0), "cm"),
    axis.text.y=element_text(size=10),
    axis.ticks.x=element_line(),
    axis.ticks.y=element_line(),
    axis.text.x = element_text(size=10,
      angle = 90,
      vjust = 0.2,
      hjust = 1),
    axis.title.y=element_text(size=10, color = "black"),
    legend.title = element_text(size=10),
    legend.text = element_text(size=10),
    legend.key=element_blank(),
    aspect.ratio = .35,
    panel.border = element_rect(size =1.5, color = "black", fill = NA),
    axis.line.y.left = element_blank(),
    axis.line.x.bottom = element_blank())+
  ylab("Pseudotime") +
  xlab("") +
  scale_y_continuous(breaks = seq(5,30,by=5)) +
  scale_x_discrete(labels = c("Q1", "Q2", "BD", "EB", "SB1", "SB2", "SB3", "SF1", "SF2"))
```

p1

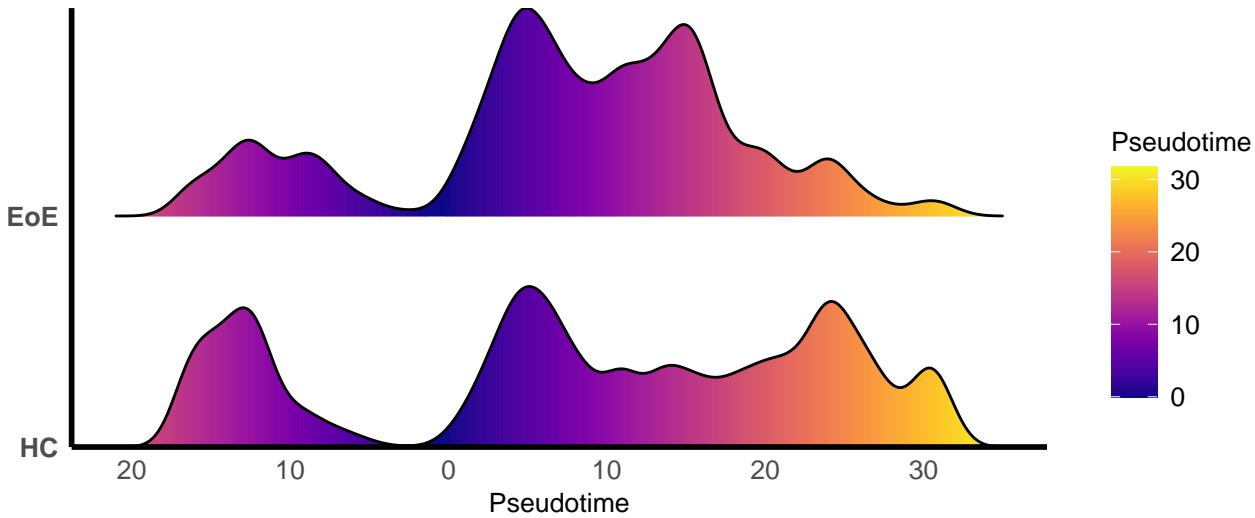


Pseudotime profile across all cells

```
# Invert quiescent cells pseudotime values to create bi-directional x axis
Pseudo_df <- Epi_Object@meta.data %>%
  mutate(DiseaseState = factor(DiseaseState, levels = c("EoE_Biopsy", "Healthy_Control")),
         Pseudotime = ifelse(Clusters %in% levels(Epi_Object@meta.data$Clusters)[1:2],
                           Pseudotime * -1,
                           as.numeric(Pseudotime)))

# Set color palette
my_palette <- c(rev(viridis::viridis(32, option = "C"))[1:20], viridis::viridis(32, option = "C"))

# Generate ridgeline plot across each compartment for HC vs EOE
ggarrange(ggplot(Pseudo_df,
                  aes(x = Pseudotime,
                      y = DiseaseState,
                      fill = stat(x))) +
  geom_density_ridges_gradient(scale = 0.9) +
  scale_fill_gradient(name = "Pseudotime",
                      colors = my_palette) +
  xlab("Pseudotime") +
  ylab("") +
  theme(axis.line = element_line(colour = "black"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank(),
        plot.margin = unit(c(0, 0, 0, 0), "cm"),
        axis.text.y=element_text(size=10, face = "bold"),
        axis.ticks.x=element_blank(),
        axis.ticks.y=element_blank(),
        axis.text.x=element_text(size=10),
        axis.title.x=element_text(size=10, color = "black"),
        plot.title = element_text(size=10),
        legend.title = element_text(size=10),
        legend.text = element_text(size=10),
        legend.key= element_blank(),
        axis.line.x.bottom = element_line(color = "black", size =1),
        axis.line.y.left = element_line(color = "black", size =1),
        aspect.ratio = .45,
        legend.position = "none") +
  scale_y_discrete(limits = c("Healthy_Control", "EoE_Biopsy"),
                   expand = c(0,0),
                   labels = c("HC", "EoE")) +
  scale_x_continuous(limits = c(-21,35),
                     breaks = seq(-20, 30, by = 10),
                     labels = c(seq(20,0,by=-10),seq(10,30,by=10))),
  get_legend(p1),
  ncol = 2,
  widths = c(4, 1))
```



Pseudotime profile across epithelial compartments

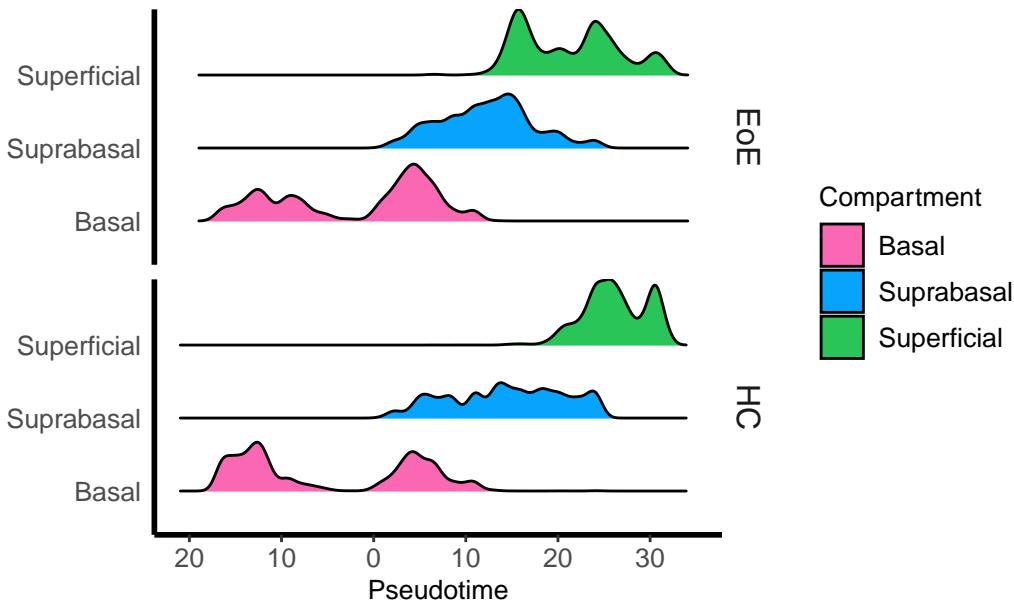
```
# Invert quiescent cells pseudotime values to create bi-directional x axis
Pseudo_df <- Epi_Object@meta.data %>%
  mutate(DiseaseState = factor(DiseaseState, levels = c("EoE_Biopsy", "Healthy_Control")),
         Pseudotime = ifelse(Clusters %in% levels(Epi_Object@meta.data$Clusters)[1:2],
                           Pseudotime * -1,
                           as.numeric(Pseudotime)))

# Generate ridgeline plot across all cells for HC vs EOE
ggplot(Pseudo_df,
       aes(x = Pseudotime,
           y = Compartments,
           fill = Compartments)) +
  geom_density_ridges_gradient(scale = 0.9) +
  scale_fill_manual(values = c("#FA68B4",
                               "#07A4FE",
                               "#2AC559"),
                    name = "Compartment") +
  facet_grid(DiseaseState ~ .,
             scales = "free_y",
             labeller = labeller(DiseaseState = c("Healthy_Control" = "HC", "EoE_Biopsy" = "EoE"))) +
  xlab("Pseudotime") +
  ylab("") +
  theme(axis.line = element_line(colour = "black"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
```

```

panel.background = element_blank(),
plot.margin = unit(c(0, 0, 0, 0), "cm"),
axis.text.y=element_text(size=10),
axis.ticks.y=element_blank(),
axis.text.x=element_text(size=10),
axis.title.x=element_text(size=10, color = "black"),
plot.title = element_text(size=10),
legend.title = element_text(size=10),
legend.text = element_text(size=10),
legend.key= element_blank(),
axis.line.x.bottom = element_line(color = "black", size =1),
axis.line.y.left = element_line(color = "black", size =1),
aspect.ratio =.45,
# legend.position = "none",
strip.background = element_blank(),
strip.text = element_text(size=12) +
scale_x_continuous(limits = c(-21,35),
breaks = seq(-20, 30, by = 10),
labels = c(seq(20,0,by=-10),seq(10,30,by=10)))

```



Pseudotime comparison between differentiated clusters

```

# Isolate differentiated cells

DifferentiatedCells <- Epi_Object[Epi_Object@meta.data %>% filter(Compartments %!in% levels(Epi_Object$Compartments)[1]) %>% rownames()]
DifferentiatedCells$Clusters <- factor(DifferentiatedCells$Clusters, levels = levels(Epi_Object$Clusters)[5:9])

# Plot pseudotime distribution between HC & EoE across differentiated clusters

Expression_BoxPlot(Object = DifferentiatedCells,
Feature = "Pseudotime",
ClusterIdent = "Clusters",
Category_Ident = "DiseaseState",
Cat_List = c("HC","EoE"),
my_comparisons = list(c("EoE_Biopsy", "Healthy_Control")),
colors = c("#C3D8EB", "#E0E", "#163161"), #Healthy
yTitle = "Pseudotime",
LegendTitle = "Disease State",
yhi = 35,
GroupLabels = levels(DifferentiatedCells$Clusters),
PatientLabel = "Patient_Region",
stepval = 0,
xlab_angle = 90,
font = 10,
NewNames = NULL,
ypos = 31,

```

```

hjust = 0,
test = "Wilcox",
bracket_length = 0.01,
jitter_width = 0.1,
clean_stats = TRUE)

```

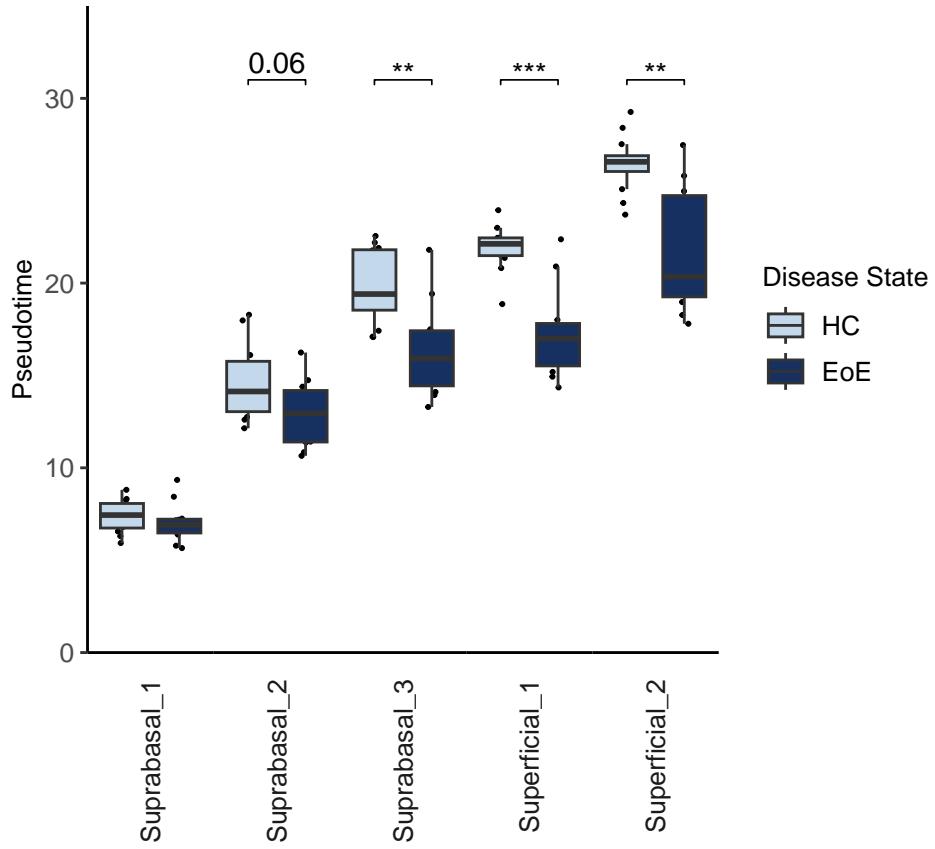


Figure 8

Module score dotplot across epithelial clusters

```

# Calculate gene signatures from pseudotime modules
setwd("/projects/b1042/MPLab/mhc0155/EoE_Project/Monocle3/EoE_Modules")
EoE_specific_pseudotime_modules <- loadRData("EoE_specific_pseudotime_modules")

for(x in 1:7){
  y <- (data.frame(EoE_specific_pseudotime_modules %>% filter(Module %in% x))$id)
  Epi_Object <- AddModuleScore(Epi_Object,
    features = list("Module" = y),
    name = paste("Module_",x,"_",sep=""))
}

# Construct dot plot across clusters

markers <- paste0("Module_",seq(1,7,by=1),"_",1,sep="")
marker(names <- paste0("Module ",seq(1,7,by=1),sep=""))

p1 <- DotPlot(Epi_Object,
  group_by = "Clusters_byDiseaseState",
  features = markers ) +
  coord_flip(clip="off") +
  theme(axis.text.x = element_text(angle=90, hjust=1, vjust = 0.5),
    plot.margin = unit(c(2,0,2,2), "lines"),
    legend.position = "top",
    legend.justification = "center",
    axis.line.x.top = element_line(),
    axis.line.y.right = element_line(),
    panel.border = element_rect(colour = "black", fill=NA, size=1),
    aspect.ratio = .5,
    legend.text = element_text(size = 9),
    legend.title = element_text(size = 9)) +
  xlab("") +

```

```

ylab("") +
scale_y_discrete(labels = rep(c("Q1", "Q2", "BD", "EB", "SB1", "SB2", "SB3", "SF1", "SF2"), 2)) +
scale_x_discrete(labels = markersnames) +
scale_colour_gradient2(low="#382C84", mid="lightgrey", high="#EF0053")

# Annotate plot

## Generate grobs for disease state labels
text1 <- grid::textGrob("HC",
  x = (0.25),
  y =(.54),
  hjust = 0.5,
  gp = gpar(cex = 1,
            fontface="bold"))
line1 <- grid::linesGrob(unit(c(0.0, 0.49), "npc"),
  unit(c(.57, .57), "npc"),
  gp = gpar(lwd = 2))

text2 <- grid::textGrob("EoE",
  x = (0.75),
  y =(.54),
  hjust = 0.5,
  gp = gpar(cex = 1,
            fontface="bold"))
line2 <- grid::linesGrob(unit(c(0.51, 1), "npc"),
  unit(c(.57, .57), "npc"),
  gp = gpar(lwd = 2))

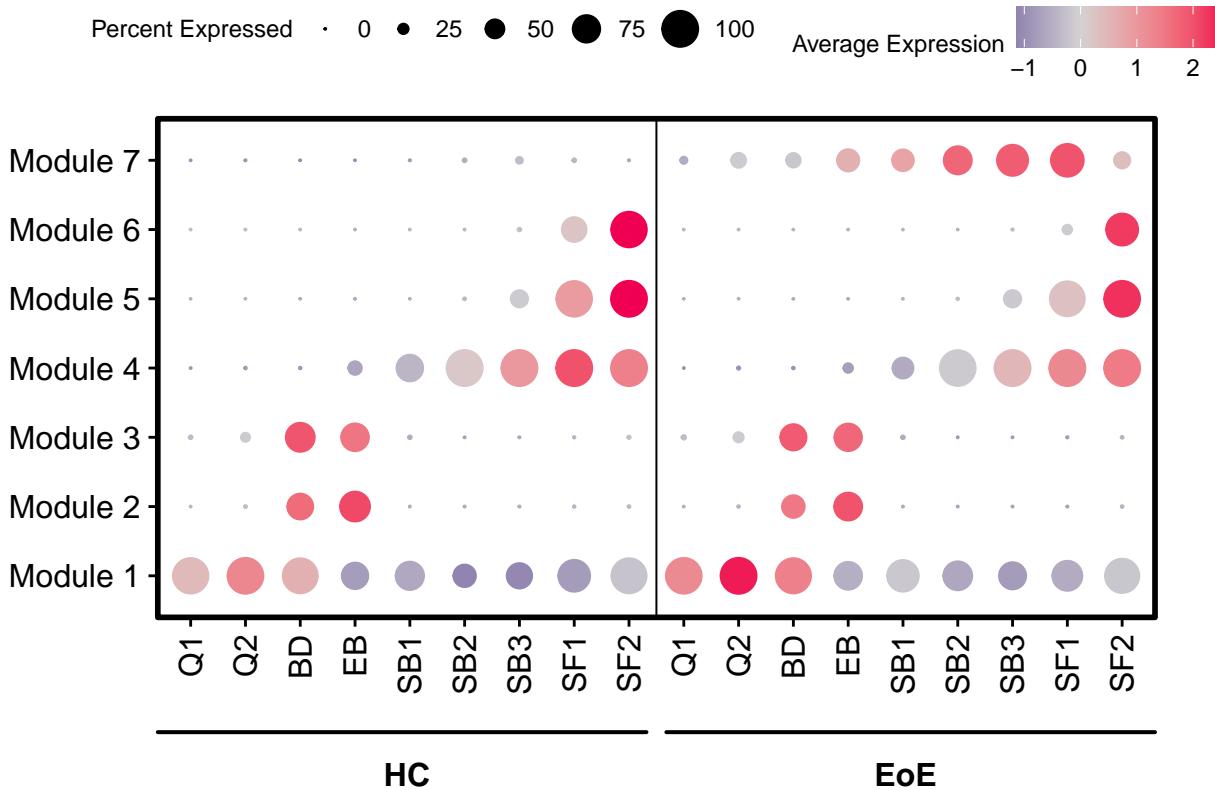
combo <- grid::grobTree(text1,line1,text2,line2)

## Attach annotation to plot
p2 <- p1 + annotation_custom(combo,
  xmin = -13,
  xmax = Inf,
  ymin= -Inf,
  ymax= Inf)

## Divide plot area for readability
gline_vert <- linesGrob(y=0:1,
  x=.5,
  gp=gpar(cex=4))
p3 <- p2 + annotation_custom(gline_vert,
  xmin = -Inf,
  xmax = Inf,
  ymin = -Inf,
  ymax = Inf)

## Display annotated plot
p3

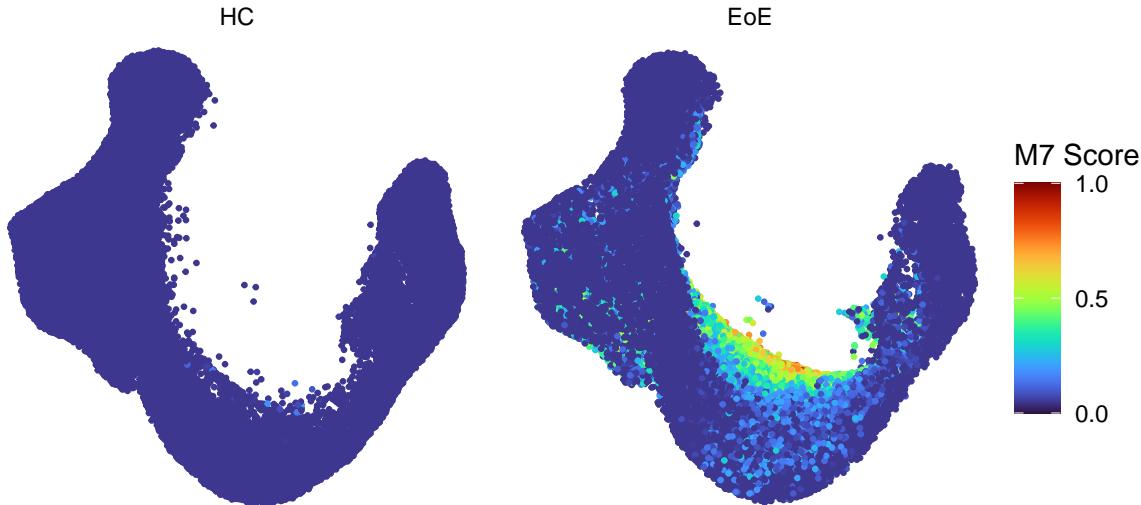
```



Module 7 scoring UMAP visualization

```
# Generate df
Module7_df <- left_join(data.frame(Epi_Object_Merged@reductions$umap@cell.embeddings) %>% rownames_to_column(var = "Cells"),
                         data.frame(Epi_Object_Merged@meta.data %>% select(DiseaseState, Module_7_1)) %>% rownames_to_column(var = "Cells"))

# Generate FeaturePlot colored by Module 7 gene signature
ggplot(Module7_df,
       aes(x = UMAP_1, y = UMAP_2, color = Module_7_1)) +
  geom_point(size = 0.5) +
  facet_wrap("DiseaseState",
             labeller = as_labeller(c(Healthy_Control = "HC",
                                      EoE_Biopsy = "EoE"))) +
  scale_color_viridis(option = "turbo",
                      limits = c(0,1),
                      breaks = seq(0,1,by=0.5),
                      name = "M7 Score") +
  labs(color = NULL) +
  theme_void() +
  theme(plot.margin = unit(c(0, 0, 0, 0), "cm"),
        axis.text.y = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.ticks.y = element_blank(),
        legend.position = "right",
        aspect.ratio=1)
```



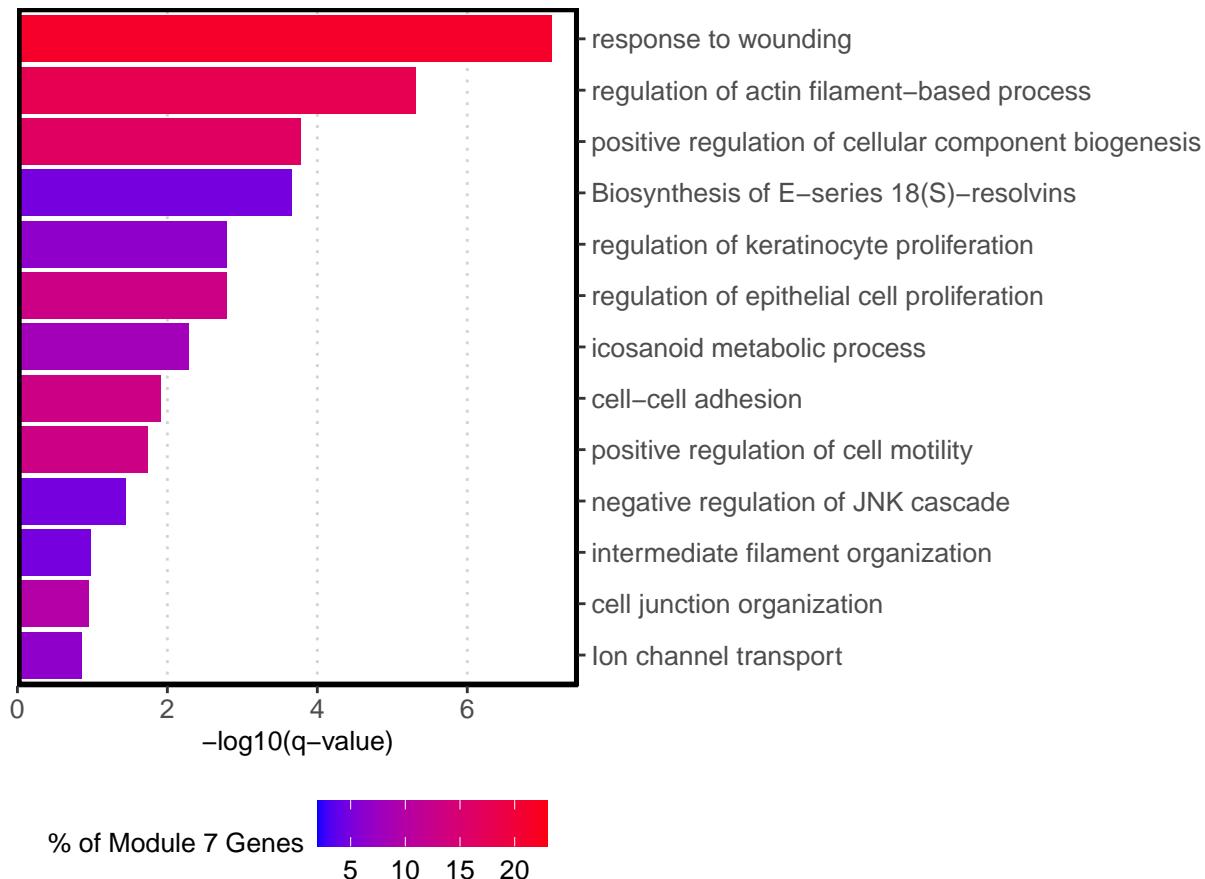
Module 7 pathway enrichment analysis visualization

```
# Read in pathway enrichment from Metascape
setwd("/projects/b1042/MPLab/mhc0155/EoE_Project/Monocle3/EoE_Modules")
M7_metascape <- loadRData("Module7_Metascape_Results.csv")

M7_metascape_df <- M7_metascape %>%
  arrange(desc(LogP)) %>%
  dplyr::slice_head(n = 13) %>%
  mutate(Description = factor>Description, levels = .$Description)) %>%
  mutate(PercentModule = 100* X.GeneInGOAndHitList/X.GeneInHitList)

M7_metascape_df %>%
  ggplot(aes(x=Description,y=-Log.q.value.,fill=PercentModule))+ 
  geom_bar(stat="identity") +
  coord_flip() +
  theme_bw() +
  theme(panel.grid.major.x = element_line(color = "lightgrey",linetype='dotted'),
        panel.grid.minor.x = element_blank(),
        panel.grid.major.y = element_blank(),
        axis.text.x = element_text(size =10),
        axis.text.y = element_text(size =10),
        legend.title = element_text(size=10),
        legend.text = element_text(size=10),
        panel.border = element_rect(color = "black", size = 1.5),
        axis.title.x = element_text(size=10),
        plot.title = element_text(size=10),
        axis.line.x.bottom = element_line(color = "black"),
        axis.line.y.left = element_line(color = "black"),
        legend.position = "bottom") +
  xlab("")+
  ylab("-log10(q-value)")+
  scale_fill_gradientn(colours = rev(c("#FD0010","#B100AA","#0000FF")),
                       values = rescale(c(min(M7_metascape_df$PercentModule),
                                         mean(M7_metascape_df$PercentModule),
                                         max(M7_metascape_df$PercentModule))),
                       guide = "colorbar", limits=c(2,23),
                       breaks = seq(5,20,by=5),
                       name = "% of Module 7 Genes") +
  scale_y_continuous(expand = c(0, 0), limits = c(0, -1.05*min(M7_metascape_df$Log.q.value.))) +
  scale_x_discrete(position = "top")+
  ggtitle("Enriched Pathways: Module 7")
```

Enriched Pathways: Module 7



SOX2, KLF5, Module 7 gene signature, and overlap expression across pseudotime

```
# Gather expression, pseudotime, and gene signature data
Expression_df <- FetchData(Epi_Object,
  vars = c("Pseudotime",
          "Clusters",
          "DiseaseState",
          "SOX2",
          "KLF5",
          "Module_7_1"),
  assay = "RNA",
  slot = "data") %>%
  mutate(Pseudotime = ifelse(Clusters %in% levels(Epi_Object_Merged$Clusters)[1:2], as.numeric(Pseudotime), as.numeric(Pseudotime)))

# Determine positive expression minimum bounds to calculate binary expression overlap columns
Expression_df <- Expression_df %>%
  mutate(Overlap_SOX2_KLF5 = ifelse((SOX2 > quantile((FetchData(Epi_Object,
    assay = "RNA",
    slot = "data",
    vars = "SOX2")) %>%
    filter(row.names(.) %in% (Epi_Object@meta.data %>%
      filter(DiseaseState %in% "Healthy_Control",
      Clusters %in% levels(Epi_Object$Clusters)[1:2])) %>%
      row.names()), SOX2 > 0.001))$SOX2,
    probs = 0.15)) &
  KLF5 > quantile((FetchData(Epi_Object,
    assay = "RNA",
    slot = "data",
    vars = "KLF5")) %>%
    filter(row.names(.) %in% (Epi_Object@meta.data %>%
      filter(DiseaseState %in% "Healthy_Control",
      Clusters %in% levels(Epi_Object$Clusters)[5])) %>%
      row.names()), KLF5 > 0.001))$KLF5,
    probs = 0.15), 1, 0),
  Overlap_All = ifelse(Overlap_SOX2_KLF5 == 1 &
    Module_7_1 > quantile((FetchData(Epi_Object,
      assay = "RNA",
      slot = "data",
      vars = "Module_7_1")) %>%
      filter(row.names(.) %in% (Epi_Object@meta.data %>%
        filter(DiseaseState %in% "EoE_Biopsy")) %>%
        row.names()))$Module_7_1,
      probs = 0.15), 1, 0))
```

```

# Function to generate local average plots over pseudotime

LocalAveragePlot <- function(df,
                           xVar,
                           yVar,
                           colorVar,
                           span,
                           yAxisTitle,
                           xAisTitle,
                           plotTitle){

p1 <- df %>%
  ggplot(aes(x=!!as.name(xVar),
             y=!!as.name(yVar),
             color = !!as.name(colorVar))) +
  geom_point(alpha=0) +
  labs(x= paste(xAisTitle), y= paste(yAxisTitle)) +
  geom_smooth(se=FALSE,
              method = "loess",
              size = 1,
              span = span) +
  scale_color_manual(values = c("#163161", "#C3D8EB"),
                     labels = c("HC", "EoE"),
                     name = "Disease State")+
  theme(axis.line = element_line(colour = "black"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        plot.margin = unit(c(0, 0, 0, 0), "cm"),
        axis.text.y=element_text(size=10),
        axis.ticks.x=element_line(),
        axis.ticks.y=element_line(),
        axis.text.x = element_text(size=10),
        axis.title.x=element_text(size=10, color = "black"),
        axis.title.y=element_text(size=10, color = "black"),
        legend.title = element_text(size=10),
        legend.text = element_text(size=10),
        legend.key=element_blank(),
        aspect.ratio = 1,
        panel.border = element_rect(size =1.5, color = "black", fill = NA),
        axis.line.y.left = element_blank(),
        axis.line.x.bottom = element_blank(),
        plot.title = element_text(size = 10, face = "bold", hjust = 0.5)) +
  ggtitle(paste(plotTitle)) +
  scale_x_continuous(limits = c(-21,32),
                     breaks = seq(-20, 30, by = 10),
                     labels = c(seq(20,0,by=-10),seq(10,30,by=10)))

  return(p1)
}

LocalAveragePlot <- function(df,
                           xVar,
                           yVar,
                           colorVar,
                           span,
                           yAxisTitle,
                           xAisTitle,
                           plotTitle){

p1 <- df %>%
  ggplot(aes(x=!!as.name(xVar),
             y=!!as.name(yVar),
             color = !!as.name(colorVar))) +
  geom_point(alpha=0) +
  labs(x= paste(xAisTitle), y= paste(yAxisTitle)) +
  geom_smooth(se=FALSE,
              method = "loess",
              size = 1,
              span = span) +
  scale_color_manual(values = c("#163161", "#C3D8EB"),
                     labels = c("HC", "EoE"),
                     name = "Disease State")+
  theme(axis.line = element_line(colour = "black"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        plot.margin = unit(c(0, 0, 0, 0), "cm"),
        axis.text.y=element_text(size=10),
        axis.ticks.x=element_line(),
        axis.ticks.y=element_line(),
        axis.text.x = element_text(size=10),
        axis.title.x=element_text(size=10, color = "black"),
        axis.title.y=element_text(size=10, color = "black"),
        legend.title = element_text(size=10),
        legend.text = element_text(size=10),
        legend.key=element_blank(),
        aspect.ratio = 1,
        panel.border = element_rect(size =1.5, color = "black", fill = NA),
        axis.line.y.left = element_blank(),
        axis.line.x.bottom = element_blank(),
        plot.title = element_text(size = 10, face = "bold", hjust = 0.5)) +
  ggtitle(paste(plotTitle)) +
  scale_x_continuous(limits = c(-21,32),
                     breaks = seq(-20, 30, by = 10),
                     labels = c(seq(20,0,by=-10),seq(10,30,by=10)))
}

```

```

    return(p1)
}

# Generate local average plots
spanV = 0.45

p1 <- LocalAveragePlot(df = Expression_df,
                       xVar = "Pseudotime",
                       yVar = "SOX2",
                       colorVar = "DiseaseState",
                       span = spanV,
                       yAxisTitle = "Expression",
                       xAisTitle = "",
                       plotTitle = "SOX2") +
  theme(legend.position = "none")

p2 <- LocalAveragePlot(df = Expression_df,
                       xVar = "Pseudotime",
                       yVar = "KLF5",
                       colorVar = "DiseaseState",
                       span = spanV,
                       yAxisTitle = "Expression",
                       xAisTitle = "Pseudotime",
                       plotTitle = "KLF5") +
  theme(legend.position = "none")

p3 <- LocalAveragePlot(df = Expression_df,
                       xVar = "Pseudotime",
                       yVar = "Module_7_1",
                       colorVar = "DiseaseState",
                       span = spanV,
                       yAxisTitle = "Signature Score",
                       xAisTitle = "",
                       plotTitle = "Module 7") +
  theme(legend.position = "none")

p4 <- LocalAveragePlot(df = Expression_df,
                       xVar = "Pseudotime",
                       yVar = "Overlap_All",
                       colorVar = "DiseaseState",
                       span = spanV,
                       yAxisTitle = "% Co-expression",
                       xAisTitle = "Pseudotime",
                       plotTitle = "Overlap")

```

(p1/p2) | (p3/p4)

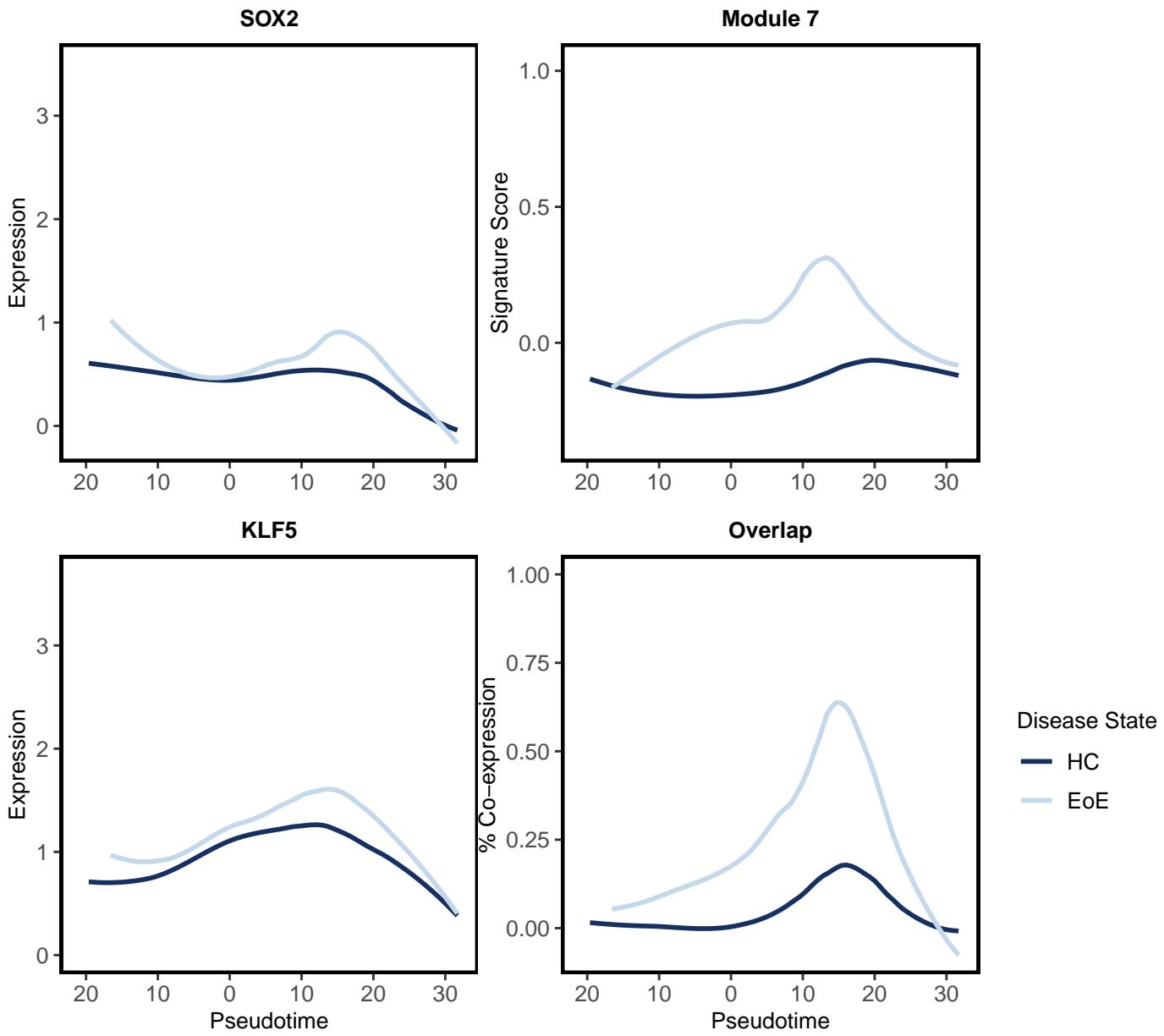


Figure 9

```
# Plot SOX2 & KLF5 expression in HC & EoE across differentiated clusters

p1 <- Expression_BoxPlot(Object = Epi_Object,
                           Feature = "SOX2",
                           ClusterIdent = "Clusters",
                           Category_Ident = "DiseaseState",
                           Cat_List = c("HC", "EoE"),
                           my_comparisons = list(c("EoE_Biopsy", "Healthy_Control")),
                           colors = c("#163161",
                                     "#C3D8EB"),
                           yTitle = "SOX2 \n Expression Level",
                           LegendTitle = "Disease State",
                           yhi = 1.45,
                           GroupLabels = levels(Epi_Object$Clusters),
                           PatientLabel = "Patient_Region",
                           stepval = 0,
                           xlab_angle = 90,
                           font = 10,
                           NewNames = NULL,
                           ypos = 1.35,
                           hjust = 0,
```

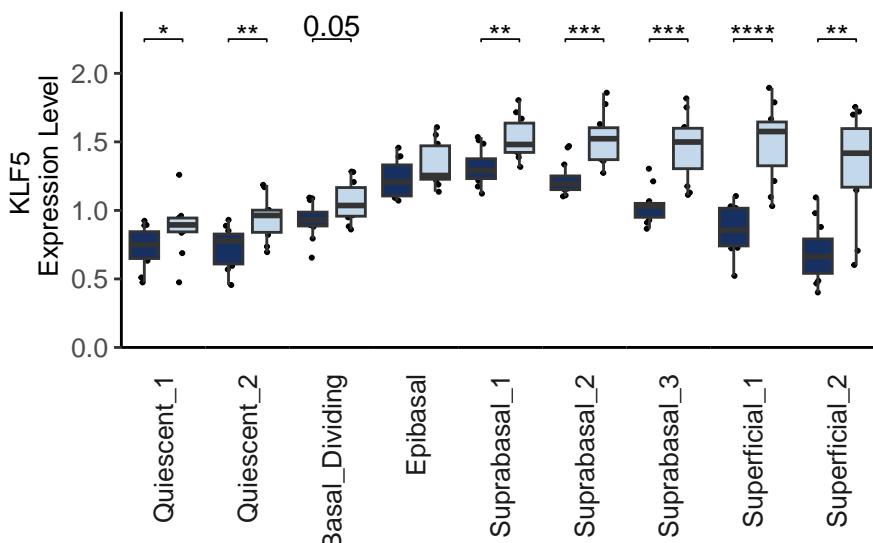
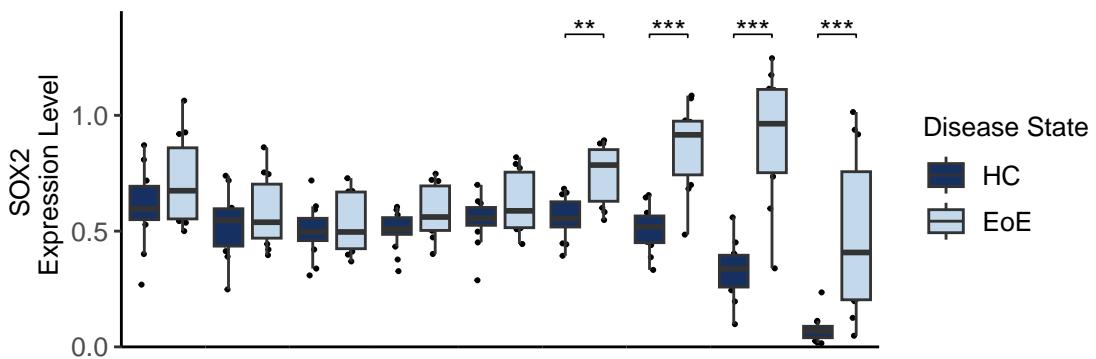
```

test = "Wilcox",
bracket_length = 0.01,
jitter_width = 0.1,
clean_stats = TRUE) +
theme(strip.text.x = element_blank())

p2 <- Expression_BoxPlot(Object = Epi_Object,
                           Feature = "KLF5",
                           ClusterIdent = "Clusters",
                           Category_Ident = "DiseaseState",
                           Cat_List = c("HC", "EoE"),
                           my_comparisons = list(c("EoE_Biopsy", "Healthy_Control")),
                           colors = c("#163161",
                                     "#C3D8EB"),
                           yTitle = "KLF5 \n Expression Level",
                           LegendTitle = "Disease State",
                           yhi = 2.45,
                           GroupLabels = levels(Epi_Object$Clusters),
                           PatientLabel = "Patient_Region",
                           stepval = 0,
                           xlab_angle = 90,
                           font = 10,
                           NewNames = NULL,
                           ypos = 2.25,
                           hjust = 0,
                           test = "Wilcox",
                           bracket_length = 0.01,
                           jitter_width = 0.1,
                           clean_stats = TRUE) +
theme(legend.position = "none")

```

p1/p2



```

# Plot SOX2 &/or KLF5 epithelial-specific gene target expression in HC & EoE across differentiated clusters
## SOX2 epithelial gene targets (PMID 30772301)
## KLF5 epithelial gene targets (PMID 34217701)
## SOX2-KLF5 epithelial gene target (PMID 33972779)

```

```

p1 <- Expression_BoxPlot(Object = Epi_Object,
                          Feature = "SOX2_EpithelialTargets",
                          ClusterIdent = "Clusters",
                          Category_Ident = "DiseaseState",
                          Cat_List = c("HC", "EoE"),
                          my_comparisons = list(c("EoE_Biopsy", "Healthy_Control")),
                          colors = c("#163161",
                                    "#C3D8EB"),
                          yTitle = "SOX2 Targets",
                          LegendTitle = "Disease State",
                          yhi = 0.17,
                          GroupLabels = levels(Epi_Object$Clusters),
                          PatientLabel = "Patient_Region",
                          stepval = 0,
                          xlab_angle = 90,
                          font = 10,
                          NewNames = NULL,
                          ypos = 0.14,
                          hjust = 0,
                          test = "Wilcox",
                          bracket_length = 0.01,
                          jitter_width = 0.1,
                          clean_stats = TRUE) +
  theme(strip.text.x = element_blank())

p2 <- Expression_BoxPlot(Object = Epi_Object,
                          Feature = "KLF5_EpithelialTargets",
                          ClusterIdent = "Clusters",
                          Category_Ident = "DiseaseState",
                          Cat_List = c("HC", "EoE"),
                          my_comparisons = list(c("EoE_Biopsy", "Healthy_Control")),
                          colors = c("#163161",
                                    "#C3D8EB"),
                          yTitle = "KLF5 Targets",
                          LegendTitle = "Disease State",
                          yhi = .45,
                          ylo = -.1,
                          GroupLabels = levels(Epi_Object$Clusters),
                          PatientLabel = "Patient_Region",
                          stepval = 0,
                          xlab_angle = 90,
                          font = 10,
                          NewNames = NULL,
                          ypos = .27,
                          hjust = 0,
                          test = "Wilcox",
                          bracket_length = 0.01,
                          jitter_width = 0.1,
                          clean_stats = TRUE) +
  theme(legend.position = "none",
        strip.text.x = element_blank())

p3 <- Expression_BoxPlot(Object = Epi_Object,
                          Feature = "SOX2KLF5_EpithelialTargets",
                          ClusterIdent = "Clusters",
                          Category_Ident = "DiseaseState",
                          Cat_List = c("HC", "EoE"),
                          my_comparisons = list(c("EoE_Biopsy", "Healthy_Control")),
                          colors = c("#163161",
                                    "#C3D8EB"),
                          yTitle = "SOX2-KLF5 Targets",
                          LegendTitle = "Disease State",
                          yhi = .16,
                          GroupLabels = levels(Epi_Object$Clusters),
                          PatientLabel = "Patient_Region",
                          stepval = 0,
                          xlab_angle = 90,
                          font = 10,
                          NewNames = NULL,
                          ypos = .14,
                          hjust = 0,
                          test = "Wilcox",
                          bracket_length = 0.01,
                          jitter_width = 0.1,
                          clean_stats = TRUE) +
  theme(legend.position = "none")

```

p1/p2/p3

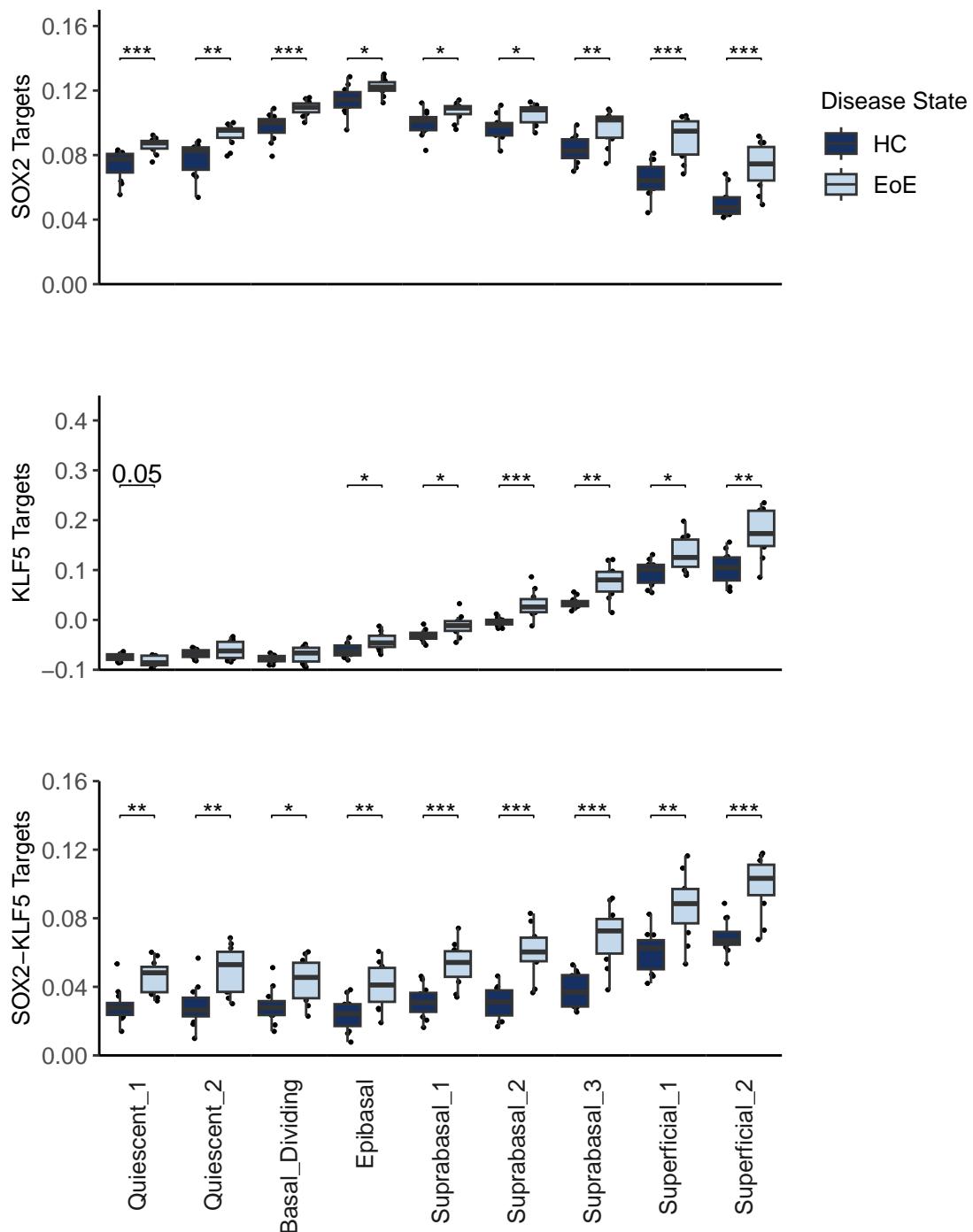


Figure 10

EoE DEG expression in EoE vs GERD as compared to HC

```
# Load in EoE DEGs per compartment
setwd("/projects/b1042/MPLab/mhc0155/EoE_Project/DEG/Compartments")
B_DEGs <- loadRData("Basal_Compartment_DEG_EoE_to_HC")
SB_DEGs <- loadRData("Suprabasal_Compartment_DEG_EoE_to_HC")
SF_DEGs <- loadRData("Superficial_Compartment_DEG_EoE_to_HC")

# Function to calculate logFC in EoE vs GERD
#' LogFC_Summarizer
#'
#' @param DEGs input list of EoE DEGs
#' @param Seurat object Seurat object containing EoE & GERD & HC
```

```

#' @param LogFC Lower logFC cutoff for input EoE DEGs
#' @param Cluster Cluster/Compartment to analyze (DEG source)
#' @param Cluster_Ident metadata column for cluster/compartment
#' @param Category_Ident metadata column for disease state annotation
#' @param PatientLabel metadata column for patient/region annotation
#'
#' @return Scatter plot comparing the logFC of EoE DEGs from HC in GERD vs EoE, by compartment
#'
#'
LogFC_Summarizer <- function(DEGs,
                               Object,
                               LogFC = 0.5,
                               Cluster,
                               Cluster_Ident = "Compartments",
                               Category_Ident = "DiseaseState",
                               PatientLabel = "Patient_Region"){

  # Filter EoE DEG list for FDR and logFC

  FilteredGenes <- DEGs %>%
    dplyr::filter(FDR_Pval < 0.05,
                  abs(Log2FC) > LogFC) %>%
    mutate(Direction = ifelse(Log2FC > LogFC, "up", "down"))

  # Calculate average expression of filtered DEGs across patients

  Object$CombCol <- paste0(Object@meta.data[[Category_Ident]],
                            ".",
                            Object@meta.data[[PatientLabel]],
                            ".",
                            Object@meta.data[[Cluster_Ident]])

  Exp_df <- AverageExpression(Object,
                                features = row.names(FilteredGenes),
                                assay = "RNA",
                                slot = "data",
                                group.by = "CombCol")

  Patient_Exp <- pivot_longer(data.frame(Exp_df$RNA) %>% mutate(gene = row.names(Exp_df$RNA)),
                               cols = colnames(data.frame(Exp_df$RNA)),
                               names_to = "Group_Summary",
                               values_to = "Avg_Expression") %>%
    separate(Group_Summary, into = c(Category_Ident, PatientLabel, Cluster_Ident), sep = "\\.") %>%
    filter(!as.name(Cluster_Ident) %in% Cluster) %>%
    mutate(!as.name(Category_Ident) := factor(!as.name(Category_Ident), levels = levels(Object@meta.data[[Category_Ident]]))) %>%
    left_join(., FilteredGenes %>%
      mutate(gene = row.names(.)) %>%
      dplyr::select(gene, Direction))

  # Calculate logFC per disease state from HC, across patient averages. Annotate with magnitude and direction of logFC change

  DS_Exp <- Patient_Exp %>%
    group_by(Direction, DiseaseState, gene) %>%
    summarise(AvgExp_perDisease = mean(Avg_Expression)) %>%
    pivot_wider(names_from = DiseaseState, values_from = AvgExp_perDisease) %>%
    mutate(EoE_LogFC = log2(EoE_Biopsy / Healthy_Control),
          GERD_LogFC = log2(GERD / Healthy_Control),
          Compartment = Cluster) %>%
    dplyr::select(Compartment, gene, Direction, EoE_LogFC, GERD_LogFC) %>%
    mutate(Color = ifelse(EoE_LogFC > LogFC & GERD_LogFC > LogFC, paste("Changed in the same direction, abs(logFC) > ", LogFC),
                           ifelse(EoE_LogFC < -LogFC & GERD_LogFC < -LogFC, paste("Changed in the same direction, abs(logFC) > ", LogFC),
                                 ifelse(EoE_LogFC < -LogFC & GERD_LogFC > LogFC, paste("Decreased < -", LogFC, "only in EoE"),
                                       ifelse(EoE_LogFC > LogFC & GERD_LogFC < LogFC & GERD_LogFC > -LogFC, paste("Increased > ", LogFC, "only in EoE"),
                                         ifelse(EoE_LogFC > LogFC & GERD_LogFC < LogFC & GERD_LogFC == 0, paste("Increased > ", LogFC, "only in EoE"),
                                               ifelse(EoE_LogFC > LogFC & GERD_LogFC < -LogFC, "Changed in the opposite direction",
                                                 ifelse(EoE_LogFC < -LogFC & GERD_LogFC > LogFC, "Changed in the opposite direction",
                                                       ""))))))) %>%
    mutate(Color = factor(Color, levels = c(paste("Increased > ", LogFC, "only in EoE"),
                                             paste("Decreased < -", LogFC, "only in EoE"),
                                             paste("Changed in the same direction, abs(logFC) > ", LogFC),
                                             "Changed in the opposite direction")))

  # Construct scatter plot comparing logFCs

  ThemePlot_Scatter <- theme(panel.background = element_rect(color = "white", fill = "white"),
                             panel.grid.major.x = element_blank(),
                             panel.grid.minor.x = element_blank(),
                             panel.grid.major.y = element_blank(),
                             panel.grid.minor.y = element_blank(),
                             panel.border = element_rect(color = "black", fill = NA),
                             aspect.ratio = 1,
                             plot.title = element_text(hjust=0.5),
                             legend.key=element_blank(),
                             legend.text = element_text(size = 8),
                             legend.title = element_text(size = 10),
                             axis.title = element_text(size = 8),
                             axis.text = element_text(size = 8))

  DEG_Scatter <- ggplot(DS_Exp, aes(x=GERD_LogFC, y=EoE_LogFC)) +
    geom_point(aes( color=Color), size = 0.25) +
    ggtitle(paste0(Cluster, " DEGs: from EoE vs HC")) +
    xlab("GERD log2FC from HC") + ylab("EoE log2FC from HC") +
    scale_color_manual(values = c("red","blue", "#01744B", "#CB00B9"),

```

```

    name = "DEG: direction of change") +
  # bold the axes
  geom_hline(yintercept = 0, color = "black") +
  geom_vline(xintercept = 0, color = "black") +
  
  ## Set theme and bounds
  ThemePlot_Scatter +
  scale_y_continuous(limits = c(-6,6), breaks = seq(-6,6,by=2)) +
  scale_x_continuous(limits = c(-5,5), breaks = seq(-4,4,by=2)) +
  
  ## calculate correlation coefficient & significance
  geom_smooth(method = "lm", color = "black", linetype = "dashed") +
  stat_cor(p.accuracy = 0.001, r.accuracy = 0.01, size = 2.5) +
  guides(colour = guide_legend(override.aes = list(size=4))) +
  
  ## Add dashed intercepts
  geom_hline(yintercept = 0.5, color = "grey", linetype = "dashed", size = 0.45) +
  geom_vline(xintercept = 0.5, color = "grey", linetype = "dashed", size = 0.45) +
  geom_hline(yintercept = -0.5, color = "grey", linetype = "dashed", size = 0.45) +
  geom_vline(xintercept = -0.5, color = "grey", linetype = "dashed", size = 0.45)

# Construct pie chart summarizing logFC direction & magnitude changes

Change_Summary <- as.data.frame(table(DS_Exp$Color)) %>%
  mutate(percent = 100*(Freq / nrow(DS_Exp))) %>%
  arrange(match(Var1, c(paste("Increased > ", LogFC, " only in EoE"),
                        paste("Decreased < -", LogFC, "only in EoE"),
                        paste("Changed in the same direction, abs(logFC) > ", LogFC),
                        "Changed in the opposite direction")))) %>%
  mutate(colorvals = c("red","blue", "#01744B", "#CB00B9")) %>%
  arrange(desc(Freq)) %>%
  mutate(Var1 = factor(Var1, levels = .$Var1)) %>%
  mutate(caption = paste(round(percent,0), "% ", Var1, sep = ""))

DEG_Pie <- ggplot(Change_Summary, aes(x="", y=Freq, fill=Var1)) +
  geom_bar(stat="identity", width=1, color="white", size = 0.5) +
  coord_polar("y", start=0) +
  theme_void() +
  theme(legend.title = element_text(size = 10),
        legend.text = element_text(size=8),
        legend.justification = c(.6,0.8),
        aspect.ratio = 1,
        legend.position = "bottom",
        legend.direction='vertical',
        plot.title = element_text(hjust=0.5)) +
  scale_fill_manual(name = paste0(Cluster, " DEG Direction Summary"),
                    values = Change_Summary$colorvals,
                    labels = Change_Summary$caption)

# Return logFC plots and df

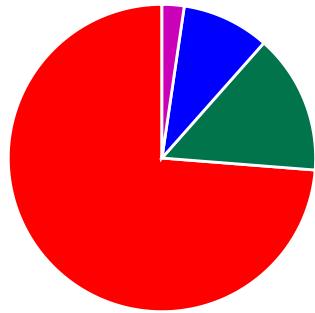
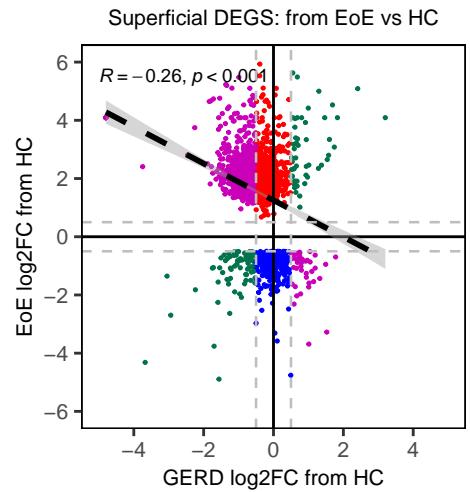
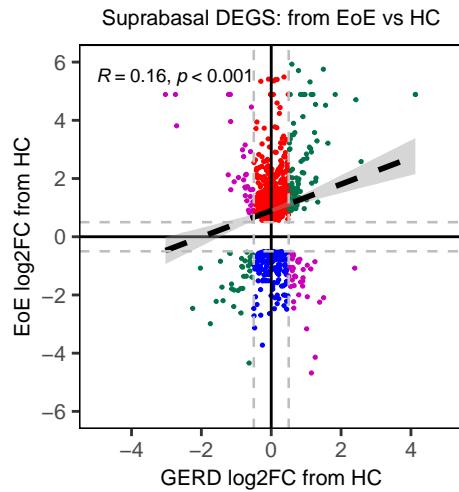
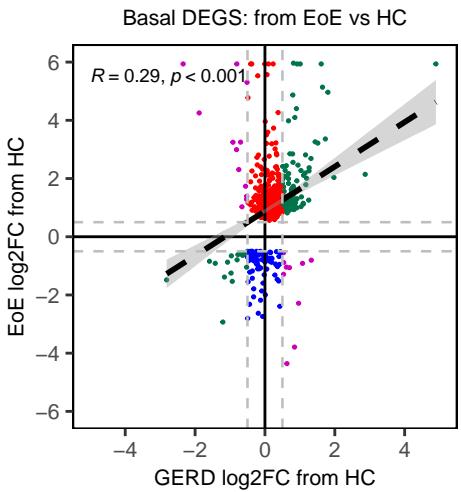
return(list(Scatter = DEG_Scatter,
            Pie = DEG_Pie,
            DEG_df = DS_Exp))
}

# Plot logFC summary across EoE and GERD compartments

B_Summary <- LogFC_Summizer(DEGs = B_DEGs,
                             Object = EoE_GERD_Epi_Object,
                             Cluster_Ident = "Compartments",
                             Cluster = "Basal",
                             LogFC = 0.5,
                             Category_Ident = "DiseaseState",
                             PatientLabel = "Patient_Region")
SB_Summary <- LogFC_Summizer(DEGs = SB_DEGs,
                             Object = EoE_GERD_Epi_Object,
                             Cluster_Ident = "Compartments",
                             Cluster = "Suprabasal",
                             LogFC = 0.5,
                             Category_Ident = "DiseaseState",
                             PatientLabel = "Patient_Region")
SF_Summary <- LogFC_Summizer(DEGs = SF_DEGs,
                             Object = EoE_GERD_Epi_Object,
                             Cluster_Ident = "Compartments",
                             Cluster = "Superficial",
                             LogFC = 0.5,
                             Category_Ident = "DiseaseState",
                             PatientLabel = "Patient_Region")

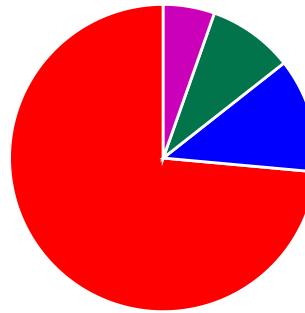
((B_Summary$Scatter + theme(legend.position = "none")) / B_Summary$Pie) |
  ((SB_Summary$Scatter + theme(legend.position = "none")) / SB_Summary$Pie) |
  ((SF_Summary$Scatter + theme(legend.position = "none")) / SF_Summary$Pie)

```



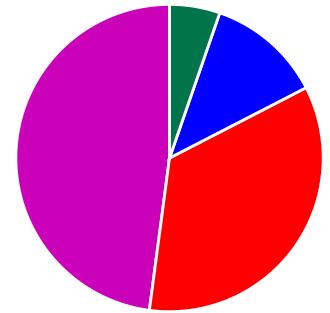
Basal DEG Direction Summary

- █ 74% Increased > 0.5 only in EoE
- █ 15% Changed in the same direction, $\text{abs}(\log FC) > 0.5$
- █ 9% Decreased < -0.5 only in EoE
- █ 2% Changed in the opposite direction



Suprabasal DEG Direction Summary

- █ 74% Increased > 0.5 only in EoE
- █ 12% Decreased < -0.5 only in EoE
- █ 9% Changed in the same direction, $\text{abs}(\log FC) > 0.5$
- █ 5% Changed in the opposite direction



Superficial DEG Direction Summary

- █ 48% Changed in the opposite direction
- █ 35% Increased > 0.5 only in EoE
- █ 12% Decreased < -0.5 only in EoE
- █ 5% Changed in the same direction, $\text{abs}(\log FC) > 0.5$

Expression of epithelial marker genes in HC vs EoE vs GERD in epithelial compartments

```
# Assign epithelial markers

markers <- c(
  # Progenitor / Basal
  "SOX2", "TP63", "KLF4", "KLF5",
  # Early Differentiation / Suprabasal
  "KRT13", "IVL",
  # Terminal Differentiation / Superficial
  "CNFN", "SPRR2D", "FLG", "KRT78"
)

# Construct dot plot

p1 <- DotPlot(EoE_GERD_Epi_Object,
  group_by = "Compartments_byDiseaseState",
  features = markers ) +
  coord_flip(clip="off") +
  theme(axis.text.x = element_text(angle=90,
    hjust=1,
    vjust = 0.5,
    size = 10),
  axis.text.y = element_text(size = 10),
  panel.border = element_rect(color = "black",
    fill = NA,
    size = 1),
  plot.margin = unit(c(2,0,2,0), "lines"),
  aspect.ratio = 1,
  legend.position = "top",
  legend.justification = "center",
  legend.title = element_text(size = 8),
  legend.text = element_text(size = 8)) +
  xlab("") +
  ylab("") +
  scale_y_discrete(labels = rep(c("B", "SB", "SF"),3)) +
  scale_colour_gradient2(low="#382C84",
    mid="lightgrey",
    high="#EF0053")
```

```

# Disease State Annotation
yval = 0.605
text1 <- grid::textGrob("HC",
  x = (0.32/2),
  y =(yval),
  hjust = 0.5,
  gp = gpar(cex = 1,
    fontface="bold"))
line1 <- grid::linesGrob(unit(c(0.0, 0.32), "npc"),
  unit(c(yval + 0.01, yval + 0.01), "npc"),
  gp = gpar(lwd = 2))

text2 <- grid::textGrob("EoE",
  x = (((0.65-0.33)/2)+0.33),
  y =(yval),
  hjust = 0.5,
  gp = gpar(cex = 1,
    fontface="bold"))
line2 <- grid::linesGrob(unit(c(0.33, 0.65), "npc"),
  unit(c(yval + 0.01, yval + 0.01), "npc"),
  gp = gpar(lwd = 2))

text3 <- grid::textGrob("GERD",
  x = (((1-0.66)/2)+0.66),
  y =(yval),
  hjust = 0.5,
  gp = gpar(cex = 1,
    fontface="bold"))
line3 <- grid::linesGrob(unit(c(0.66, 1), "npc"),
  unit(c(yval + 0.01, yval + 0.01), "npc"),
  gp = gpar(lwd = 2))

combo <- grid::grobTree(text1,line1,text2,line2,text3,line3)
p2 <- p1 + annotation_custom(combo,
  xmin = -18.5,
  xmax = Inf,
  ymin=-Inf,
  ymax=Inf)

# Annotate general plot groupings for clarity

HorizontalLine1 = linesGrob(x=0:1,
  y=.4,
  gp=gpar(cex=4,lty='longdash'))
HorizontalLine2 = linesGrob(x=0:1,
  y=.6,
  gp=gpar(cex=4,lty='longdash'))

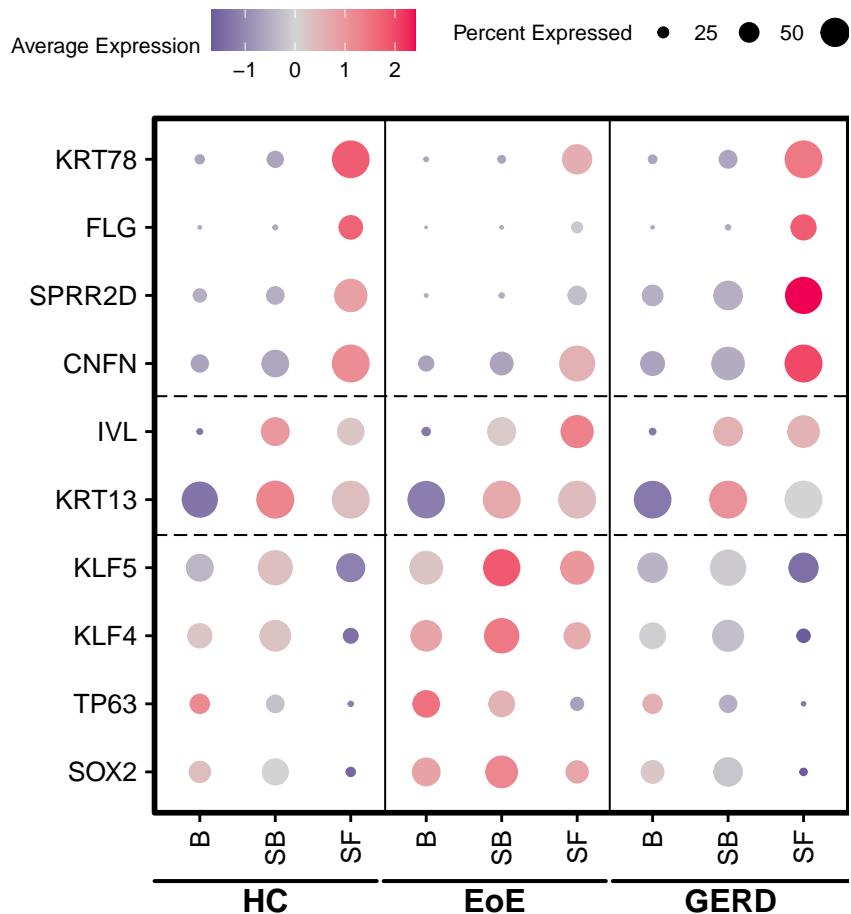
VerticalLine1 = linesGrob(y=0:1,
  x=.36,
  gp=gpar(cex=4))
VerticalLine2 = linesGrob(y=0:1,
  x=.67,
  gp=gpar(cex=4))

p3 = p2 +
  annotation_custom(HorizontalLine1,
    xmin = -Inf,
    xmax = Inf,
    ymin = -Inf,
    ymax = Inf) +
  annotation_custom(HorizontalLine2,
    xmin = -Inf,
    xmax = Inf,
    ymin = -Inf,
    ymax = Inf) +
  annotation_custom(VerticalLine1,
    xmin = -Inf,
    xmax = Inf,
    ymin = 0,
    ymax = Inf) +
  annotation_custom(VerticalLine2,
    xmin = -Inf,
    xmax = Inf,
    ymin = 0,
    ymax = Inf)

# Display dot plot

p3

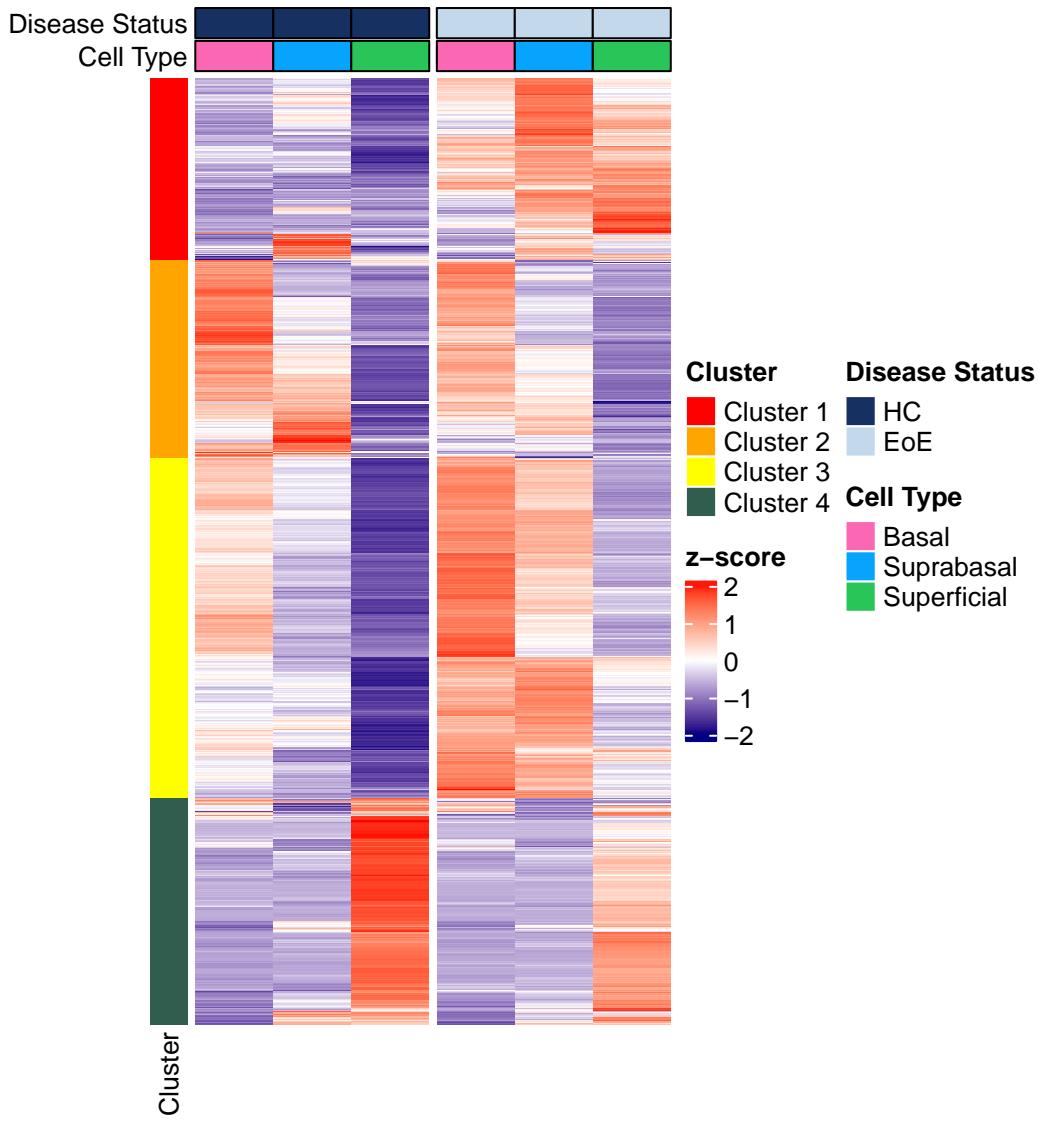
```



Expression of EoE DEG gene modules in GERD

```
#Load in heatmap from Supplemental Figure 10A
setwd("/projects/b1042/MPLab/mhc0155/EoE_Project/DEG/Suprabasal_Superficial_combined")
HM_Matrix <- loadRData("HM_DEGs_SuprabasalSuperficial_EoEtoHC_Matrix")
HM <- loadRData("HM_DEGs_SuprabasalSuperficial_EoEtoHC")

HM
```



```

# Gather genes per HM cluster
df <- HM_Matrix@matrix
rcl.list <- row_order(HM) #Extract clusters (output is a list)

geneLIST <- list()
for (i in 1:length(rcl.list)){
  clusterNum <- names(rcl.list)[i]
  if (i == 1) {
    clu <- t(t(row.names(df[rcl.list[[i]],])))
    geneLIST[[i]] <- as.vector(clu)
  } else {
    clu <- t(t(row.names(df[rcl.list[[i]],])))
    geneLIST[[i]] <- as.vector(clu)
  }
}

# Calculate gene signature for each gene cluster
for(x in 1:4){
  y <- data.frame(geneLIST[[x]])[,1]
  EoE_GERD_Epi_Object <- AddModuleScore(EoE_GERD_Epi_Object,
                                            features = list("Cluster" = y),
                                            name = paste("HM_SBSF_Cluster_",x,"_",sep=""))
}

# Construct stacked violin plot
ClusterNames <- c("HM_SBSF_Cluster_1_1",
                  "HM_SBSF_Cluster_2_1",
                  "HM_SBSF_Cluster_3_1",
                  "HM_SBSF_Cluster_4_1")
ylabs <- paste0("Cluster ", c(1:4))

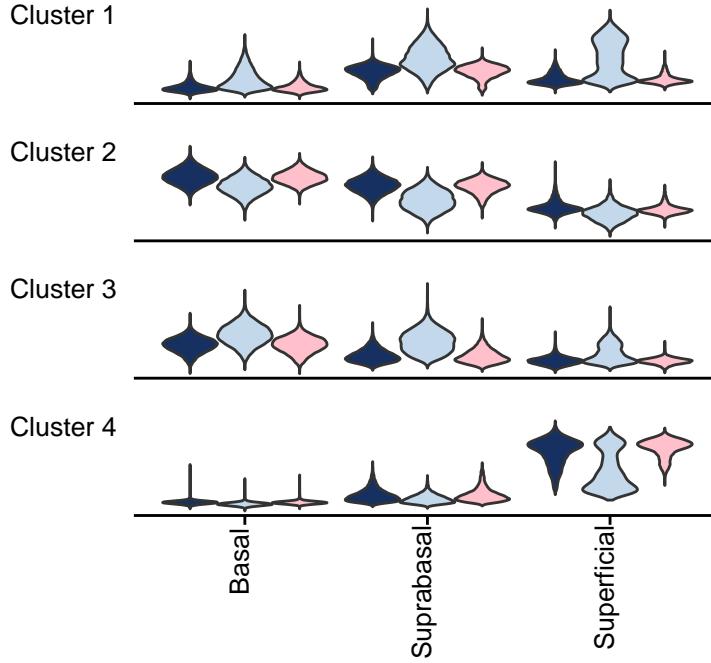
StackedVlnPlot(obj = EoE_GERD_Epi_Object,
               features = ClusterNames,
               ident = "Compartments",
               colors = c("#163161",
                         "#C3D8EB",
                         "pink"),
               )

```

```

xlab = levels(EoE_GERD_Epi_Object$Compartments),
negNum = 12.5,
split.by = "DiseaseState",
ylabs = ylabs,
font = 10)

```



Quiescent & Superficial Signature in GERD

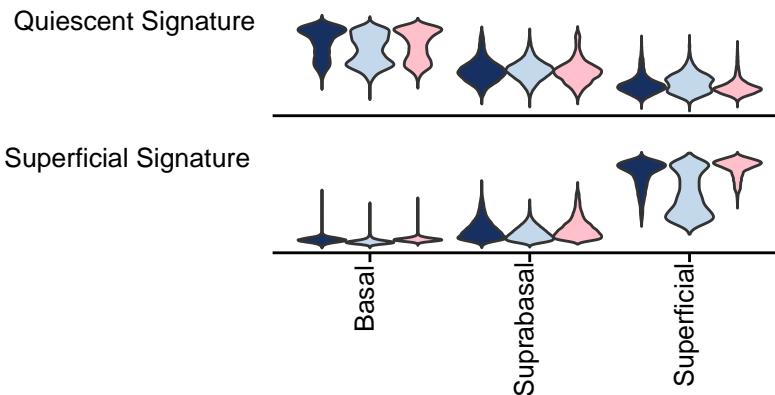
```

# Construct stacked vln plots

FeatureNames <- c("Quiescent_Sig",
                  "Superficial_Sig")
ylabs <- c("Quiescent Signature",
          "Superficial Signature")

StackedVlnPlot(obj = EoE_GERD_Epi_Object,
               features = FeatureNames,
               ident = "Compartments",
               colors = c("#163161",
                         "#C3D8EB",
                         "pink"),
               xlab = levels(EoE_GERD_Epi_Object$Compartments),
               negNum = 12.5,
               split.by = "DiseaseState",
               ylab = ylab,
               font = 10)

```



SOX2 & KLF5 expression in GERD

```
# Construct stacked vln plots
```

```
FeatureNames <- c("SOX2",
  "KLF5")

StackedVlnPlot(obj = EoE_GERD_Epi_Object,
  features = FeatureNames,
  ident = "Compartments",
  colors = c("#163161",
    "#C3D8EB",
    "pink"),
  xlabs = levels(EoE_GERD_Epi_Object$Compartments),
  negNum = 12.5,
  split.by = "DiseaseState",
  ylabs = FeatureNames,
  font = 10)
```

