SASC on CRC data

Yunshan Duan

2024-02-20

Load functions:

```
library(scry)
library(glmpca)
library(ggplot2); theme_set(theme_bw())
library(umap)
library(mvtnorm)
library(MCMCpack)
library(cluster)
library(salso)
library(ggpubr)
library(dplyr)
library(fdrtool)
library(Seurat)

source("SASC_func.R")
source("realdata_func.R")

data_dir <- "../data/realdata"
output_dir <- "../output/realdata"
output_data_dir <- "../output/realdata/data"
output_figure_dir <- "../output/realdata/figures"

# colors
pal <- c("#ffb6db", "#33A02C", "#b66dff", "#FEC44F", "#41B6C4", "#8E0152", "#0868AC", "#807DBA", "#E7298
         "#00441B", "#525252", "#4D9221", "#8B5742", "#D8DAEB", "#7cdd2d", "#980043", "#8C96C6", "#EC70
         "#FDAE61", "#1D91C0", "#A6DBA0", "#4292C6", "#BF812D", "#01665E", "#41AB5D", "#FE9929", "#2525
names(pal) <- 1:30
```

Read data:

```
input <- readRDS(paste0(data_dir, "/1_CRC_final_annotation.rds"))

pdf(file = paste0(output_figure_dir, "/PCAelbow.pdf"), width = 5, height = 4)
ElbowPlot(input, ndims = 50)
dev.off()
```

```
## pdf
##   2
```

```
npc <- 25
input <- RunPCA(object = input, npcs = npc)
```

```r
# count matrix
count <- as.matrix(input@assays$RNA@counts)
dim(count)
```

```
## [1] 12525  3139
```

```r
# feature matrix after PCA (dim reduction)
gene <- input@reductions$pca@cell.embeddings
dim(gene)
```

```
## [1] 3139   25
```

```r
# umap embeddings
gene_umap <- input@reductions$umap@cell.embeddings
dim(gene_umap)
```

```
## [1] 3139    2
```

```r
# onset
onset <- input@meta.data$Onset
# cell names
cell_names <- colnames(count)
# CD8 CD4 type
T_type <- input@meta.data$T_cell_type
table(T_type)
```

```
## T_type
##  CD4  CD8
## 1626 1513
```

```r
annotation <- input@meta.data$annotation_final
table(annotation)
```

```
## annotation
##    cd4T_hp cd4T_other    cd4T_rg    CD8T_em    CD8T_ex CD8T_other
##        858        198        570        657        362        494
```

```r
types <- names(table(annotation))
names(annotation) <- cell_names

# rename the groups to be L and E
onset_LE <- onset
onset_LE[which(onset == "LOCRC")] <- "L"
onset_LE[which(onset == "YOCRC")] <- "E"
```

Exploratory data analysis:

```r
annotation_rename <- annotation
annotation_rename[which(annotation == "cd4T_hp")] <- "CD4T_hp"
annotation_rename[which(annotation == "cd4T_other")] <- "CD4T_other"
annotation_rename[which(annotation == "cd4T_rg")] <- "CD4T_rg"
Annotation <- annotation_rename
Onset <- onset
table_df <- table(Annotation, Onset)
pdf(paste0(output_figure_dir, "/mosaic.pdf"), width=6, height = 3.5)
mosaicplot(table_df,
           main = "Mosaic plot",
```

```
           color = TRUE
)
dev.off()
```

```
## pdf
##   2
```

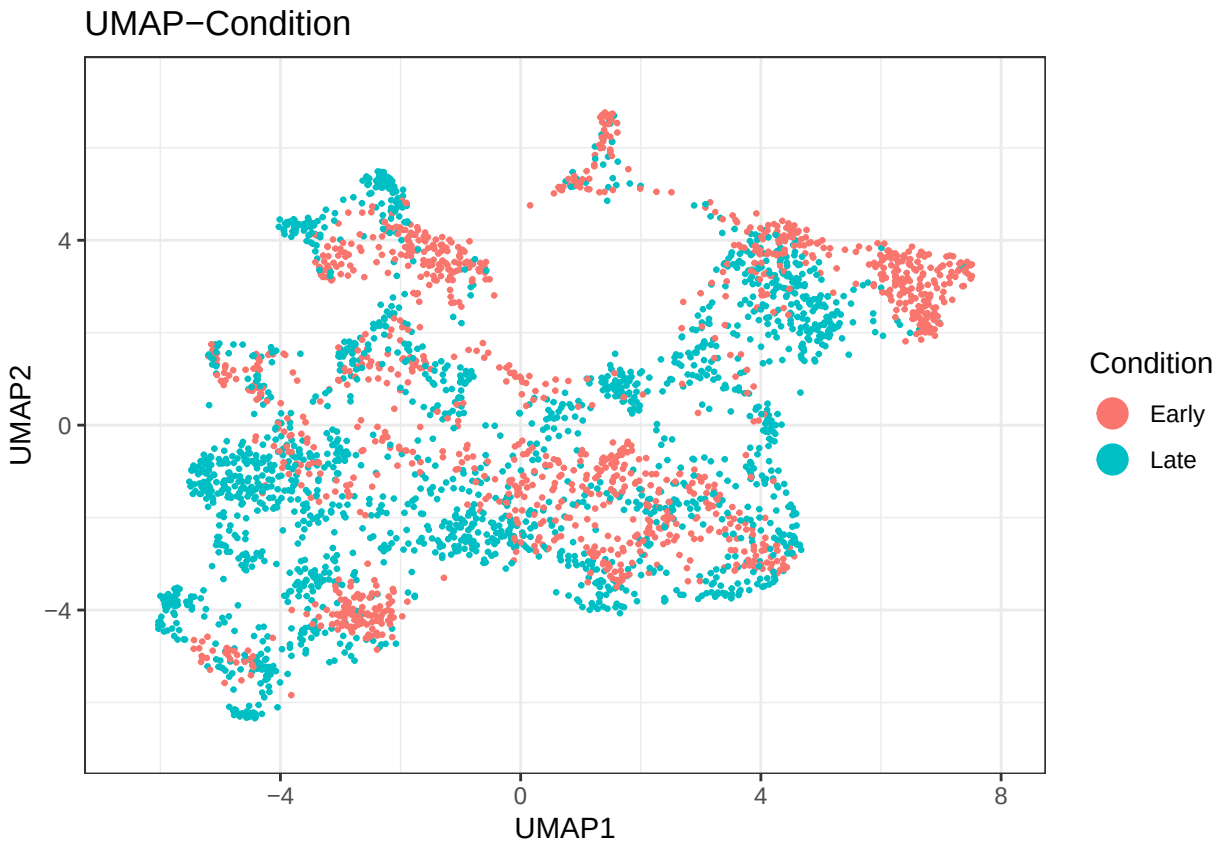
```
fisher.test(table_df, simulate.p.value=TRUE)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based on
##  2000 replicates)
##
## data:  table_df
## p-value = 0.0004998
## alternative hypothesis: two.sided
```

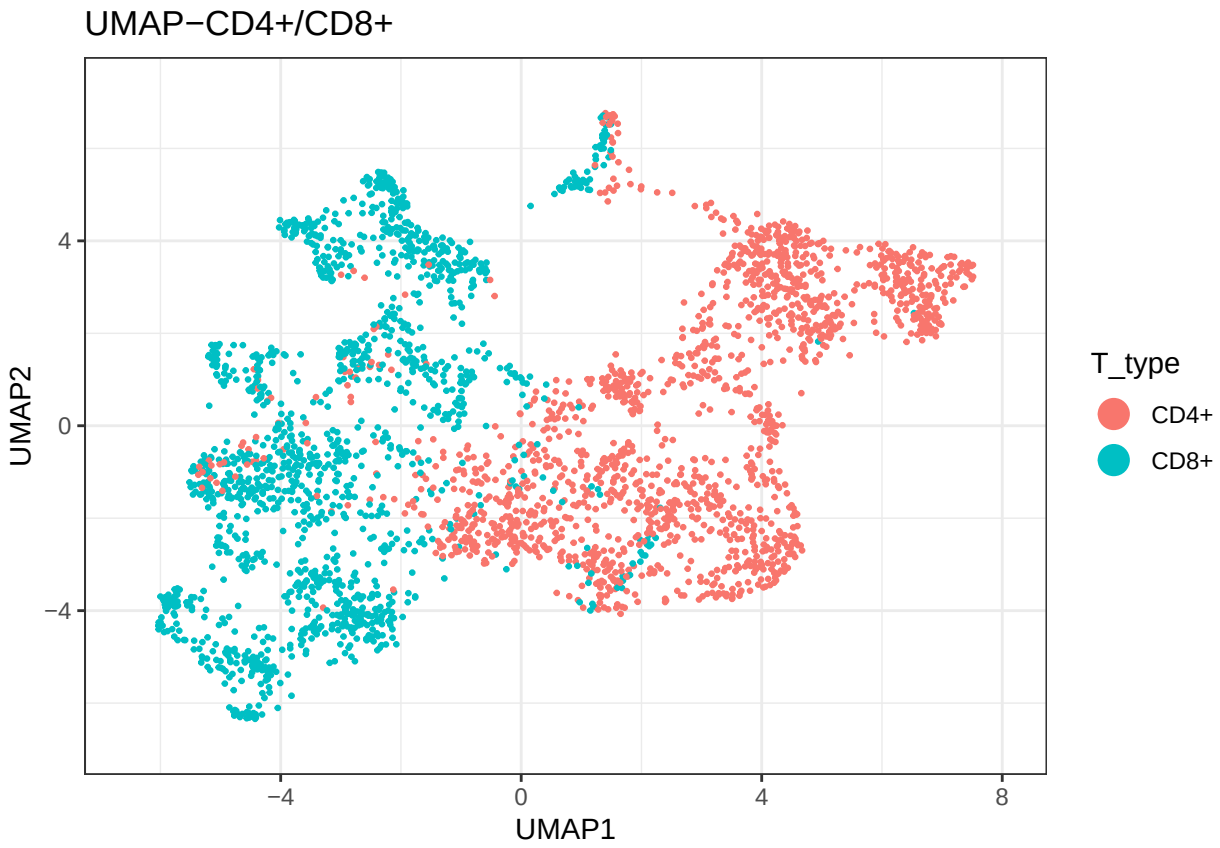
Visualize the data set:

```
pointsize <- 0.5

annotation_plot <- annotation
annotation_plot[which(annotation == "cd4T_hp")] <- "CD4+ T helper"
annotation_plot[which(annotation == "cd4T_other")] <- "CD4+ T other"
annotation_plot[which(annotation == "cd4T_rg")] <- "CD4+ T regulatory"
annotation_plot[which(annotation == "CD8T_em")] <- "CD8+ T effective memory"
annotation_plot[which(annotation == "CD8T_ex")] <- "CD8+ T exhausted"
annotation_plot[which(annotation == "CD8T_other")] <- "CD8+ T other"

pal0 <- pal[1:6]
names(pal0) <- annot_names <- c("CD4+ T helper",
                 "CD4+ T other",
                 "CD4+ T regulatory",
                 "CD8+ T effective memory",
                 "CD8+ T exhausted",
                "CD8+ T other")
T_type_plot <- T_type
T_type_plot[which(T_type == "CD4")] <- "CD4+"
T_type_plot[which(T_type == "CD8")] <- "CD8+"
onset_LE_plot <- onset_LE
onset_LE_plot[which(onset_LE == "L")] <- "Late"
onset_LE_plot[which(onset_LE == "E")] <- "Early"
df <- data.frame(PCA1 = gene[,1], PCA2 = gene[,2],
                 UMAP1 = gene_umap[,1], UMAP2 = gene_umap[,2],
                 Condition = onset_LE_plot, T_type = T_type_plot, Annotation = annotation_plot)

xrange <- c(min(df$UMAP1) - 0.5 , max(df$UMAP1) + 0.5)
yrange <- c(min(df$UMAP2) - 0.5 , max(df$UMAP2) + 0.5)

# plot
p1 <- ggplot(df, aes(x = UMAP1, y = UMAP2, color = Condition)) +
  geom_point(size = pointsize) +
  guides(colour=guide_legend(override.aes=list(size = 5))) +
  ylim(yrange) + xlim(xrange) +
```

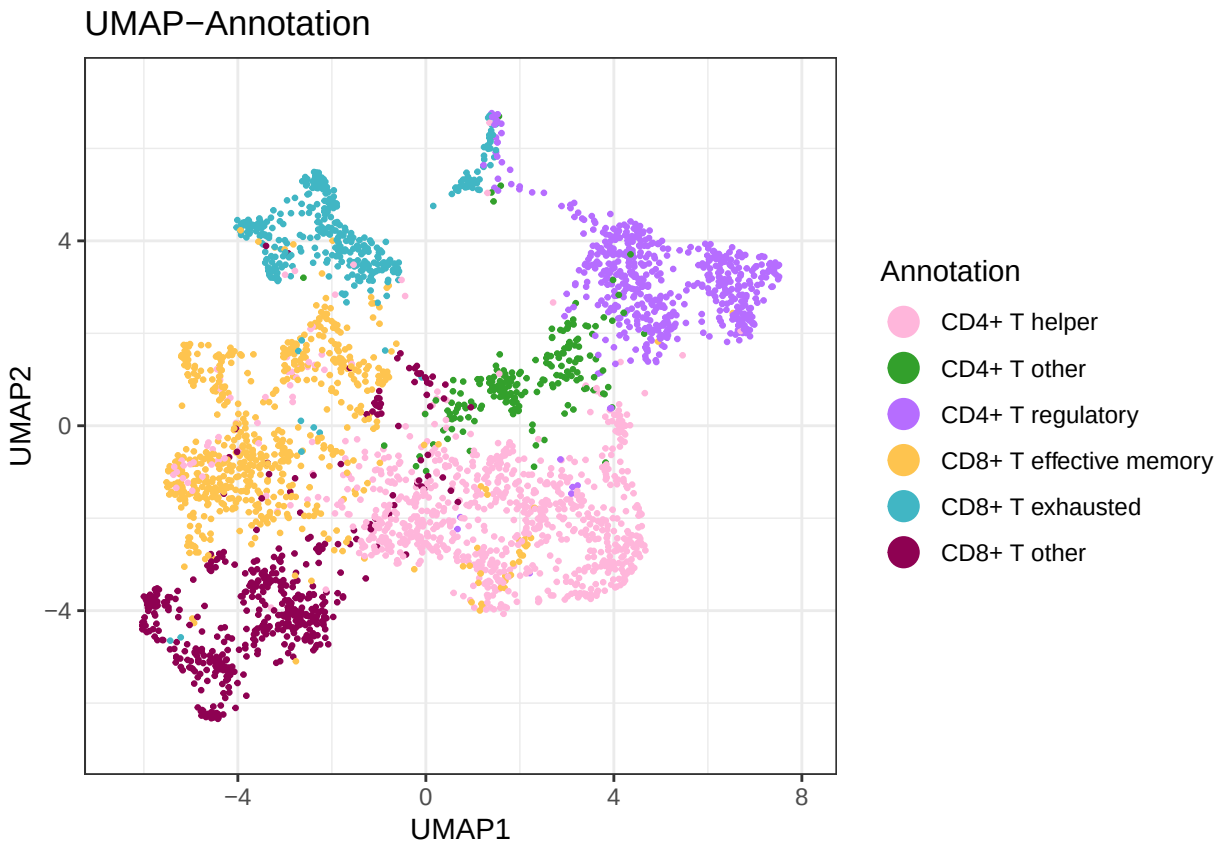
```
  ggtitle("UMAP-Condition")
p1
```



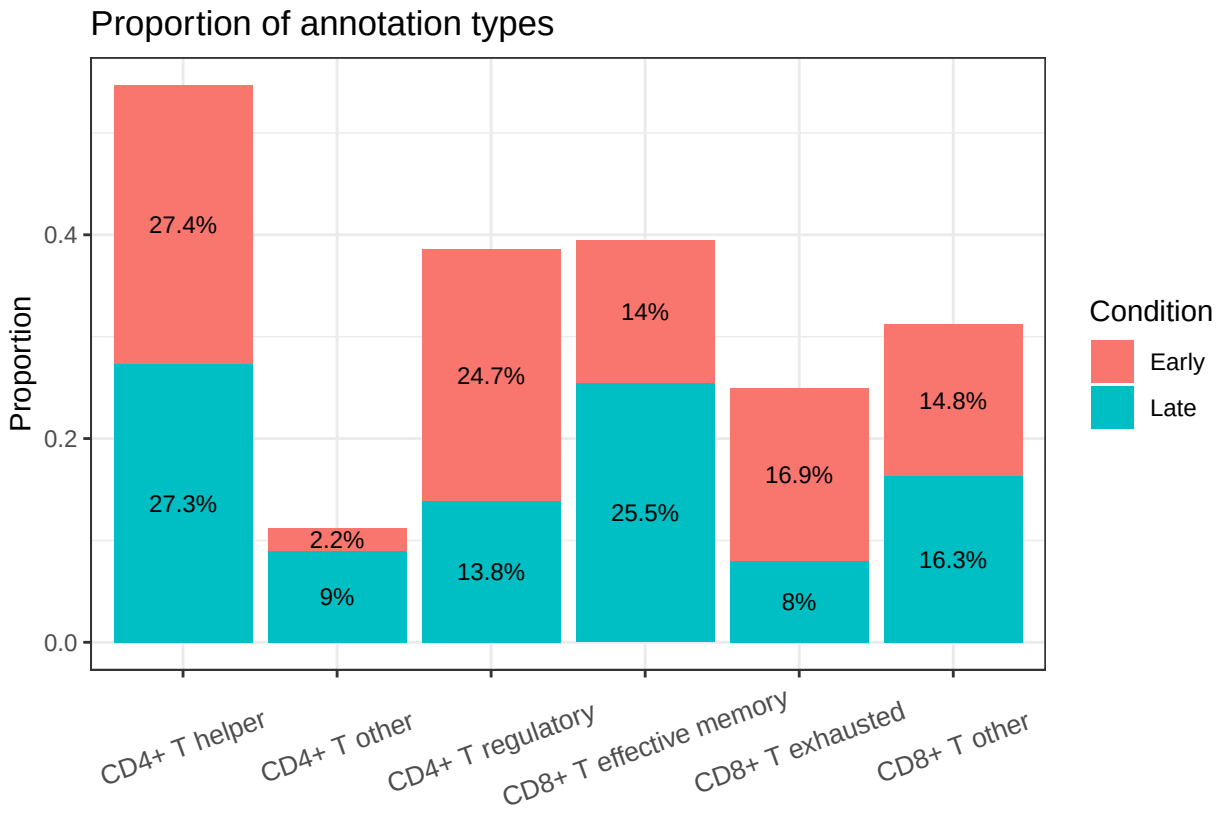

```
# plot
p2 <- ggplot(df, aes(x = UMAP1, y = UMAP2, color = T_type)) +
  geom_point(size = pointsize) +
  guides(colour=guide_legend(override.aes=list(size = 5))) +
  ylim(yrange) + xlim(xrange) +
  ggtitle("UMAP-CD4+/CD8+")
p2
```




```
p3 <- ggplot(df, aes(x = UMAP1, y = UMAP2, color = Annotation)) +
  geom_point(size = pointsize) +
  guides(colour=guide_legend(override.aes=list(size = 5))) +
  scale_color_manual(values=pal0) +
  ylim(yrange) + xlim(xrange) +
  ggtitle("UMAP-Annotation")
p3
```




```
annot_LO <- table(annotation_plot[which(onset == "LOCRC")])
annot_YO <- table(annotation_plot[which(onset == "YOCRC")])

annot_prop <- rbind(annot_LO/sum(annot_LO), annot_YO/sum(annot_YO))

n_annot <- ncol(annot_prop)
# plot
df_bar <- data.frame(Condition = c(rep("Late" , n_annot), rep("Early" , n_annot) ),
                     type = rep(colnames(annot_prop) , 2),
                     prop = c(annot_prop[1, ], annot_prop[2, ]))

p4 <- ggplot(df_bar, aes(fill=Condition, y=prop, x=type, label = paste0(round(prop, 3)*100, "%"))) +
  geom_bar(stat="identity") +
  scale_x_discrete(limits = annot_names) +
  geom_text(size = 3, position = position_stack(vjust = 0.5)) +
  ylab("Proportion") + xlab("") +
  theme(axis.text.x = element_text(angle = 20, vjust = 0.5, hjust=0.5, size = 10)) +
  ggtitle("Proportion of annotation types")
p4
```




```
plot <- ggarrange(p1, p2, ncol = 2, nrow = 1, align = "v", labels = c("(a)", "(b)"))
ggsave(filename = paste0(output_figure_dir, "/data.pdf"), width = 8, height = 3.5)

plot <- ggarrange(p3, p4, ncol = 2, nrow = 1, align = "v", labels = c("(c)", "(d)"))
ggsave(filename = paste0(output_figure_dir, "/data2.pdf"), width = 10, height = 3.5)
```

Run SASC model:

```
data_CD4 <- subset(x = input, subset = (T_cell_type == "CD4"))
out_CD4 <- run888model(data_CD4, H = 2, resolution = 0.3, fdr_thhd = 1e-60)
```

```
## Start pre-processing ...
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 1626
## Number of edges: 64097
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8504
## Number of communities: 6
## Elapsed time: 0 seconds
## Control FDR 1e-60 adding 33 biomarkers
## Finish pre-processing!
## MCMC iterations:
## 50   100     150     200     250     300     350     400     450     500     550     600     650
##  Calculating the loss of the partitions...
##
##  Got the point estimate of the partition!
```

```
##
##  Extra mcmc for estimation of the weights.
## 50   100     150     200     250     300     350     400     450     500
```

```
data_CD8 <- subset(x = input, subset = (T_cell_type == "CD8"))
out_CD8 <- run888model(data_CD8, H = 2, resolution = 0.3, fdr_thhd = 1e-60)
```

```
## Start pre-processing ...
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 1513
## Number of edges: 63221
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8560
## Number of communities: 5
## Elapsed time: 0 seconds
## Control FDR 1e-60 adding 23 biomarkers
## Finish pre-processing!
## MCMC iterations:
## 50   100     150     200     250     300     350     400     450     500     550     600     650
##  Calculating the loss of the partitions...
##
##  Got the point estimate of the partition!
##
##  Extra mcmc for estimation of the weights.
## 50   100     150     200     250     300     350     400     450     500
```

```
####
cell_names <- input@assays$RNA@counts@Dimnames[[2]]
T_type <- input@meta.data$T_cell_type
CD4_cells <- cell_names[which(T_type == "CD4")]
CD8_cells <- cell_names[which(T_type == "CD8")]

cluster <- rep(NA, length(cell_names))
names(cluster) <- cell_names
cluster[CD4_cells] <- out_CD4$cluster
n_clusters_CD4 <- max(as.numeric(out_CD4$cluster))
n_clusters_CD8 <- max(as.numeric(out_CD8$cluster))
n_clusters <- n_clusters_CD4 + n_clusters_CD8
cluster[CD8_cells] <- as.character(as.numeric(out_CD8$cluster) + n_clusters_CD4)

cluster_names <- as.character(1:n_clusters)
cluster_names
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11"
```

```
weights <- cbind(out_CD4$weights, out_CD8$weights)
cluster_type_CLE <- c(out_CD4$cluster_type_CLE, out_CD8$cluster_type_CLE)
colnames(weights) <- cluster_names

names(cluster_type_CLE) <- cluster_names

prob_in_h_mat_all <- matrix(NA, nrow = n_clusters, ncol = length(cell_names))
rownames(prob_in_h_mat_all) <- cluster_names
colnames(prob_in_h_mat_all) <- cell_names
```

```
prob_in_h_mat_all[1:n_clusters_CD4, CD4_cells] <- out_CD4$prob_in_h_mat_out
prob_in_h_mat_all[1:n_clusters_CD8, CD8_cells] <- out_CD8$prob_in_h_mat_out

xi_est <- cbind(out_CD4$xi_est[1:npc,], out_CD8$xi_est[1:npc,])
colnames(xi_est) <- cluster_names

weights_mat <- list()
weights_mat[[1]] <- cbind(out_CD4$weights_mat[[1]], out_CD8$weights_mat[[1]])
colnames(weights_mat[[1]]) <- cluster_names
weights_mat[[2]] <- cbind(out_CD4$weights_mat[[2]], out_CD8$weights_mat[[2]])
colnames(weights_mat[[2]]) <- cluster_names
```

Rename the clusters:

```
cluster_type_CLE_new <- c(cluster_type_CLE[order(match(cluster_type_CLE[1:n_clusters_CD4], c("C", "L",
                     cluster_type_CLE[(n_clusters_CD4 + 1):n_clusters][order(match(cluster_type_CLE[(n_
cluster_names_new <- names(cluster_type_CLE_new)

cluster_new <- cluster
for (i in 1:n_clusters) {
  if (as.numeric(cluster_names_new[i]) != as.numeric(cluster_names[i]) ) {
    cluster_new[which(cluster == cluster_names_new[i])] <- as.numeric(cluster_names[i])
  }
}
weights_new <- weights[, cluster_names_new]
colnames(weights_new) <- cluster_names
weights_mat_new <- list()
for (i in 1:2) {
  weights_mat_new[[i]] <- weights_mat[[i]][, cluster_names_new]
  colnames(weights_mat_new[[i]]) <- cluster_names
}
xi_est_new <- xi_est[, cluster_names_new]
colnames(xi_est_new) <- cluster_names
prob_in_h_mat_all_new <- prob_in_h_mat_all[cluster_names_new, ]
rownames(prob_in_h_mat_all_new) <- cluster_names
names(cluster_type_CLE_new) <- cluster_names

cluster <- cluster_new
weights <- weights_new
xi_est <- xi_est_new
prob_in_h_mat <- prob_in_h_mat_all_new
cluster_type_CLE <- cluster_type_CLE_new
weights_mat <- weights_mat_new
```

Save data:

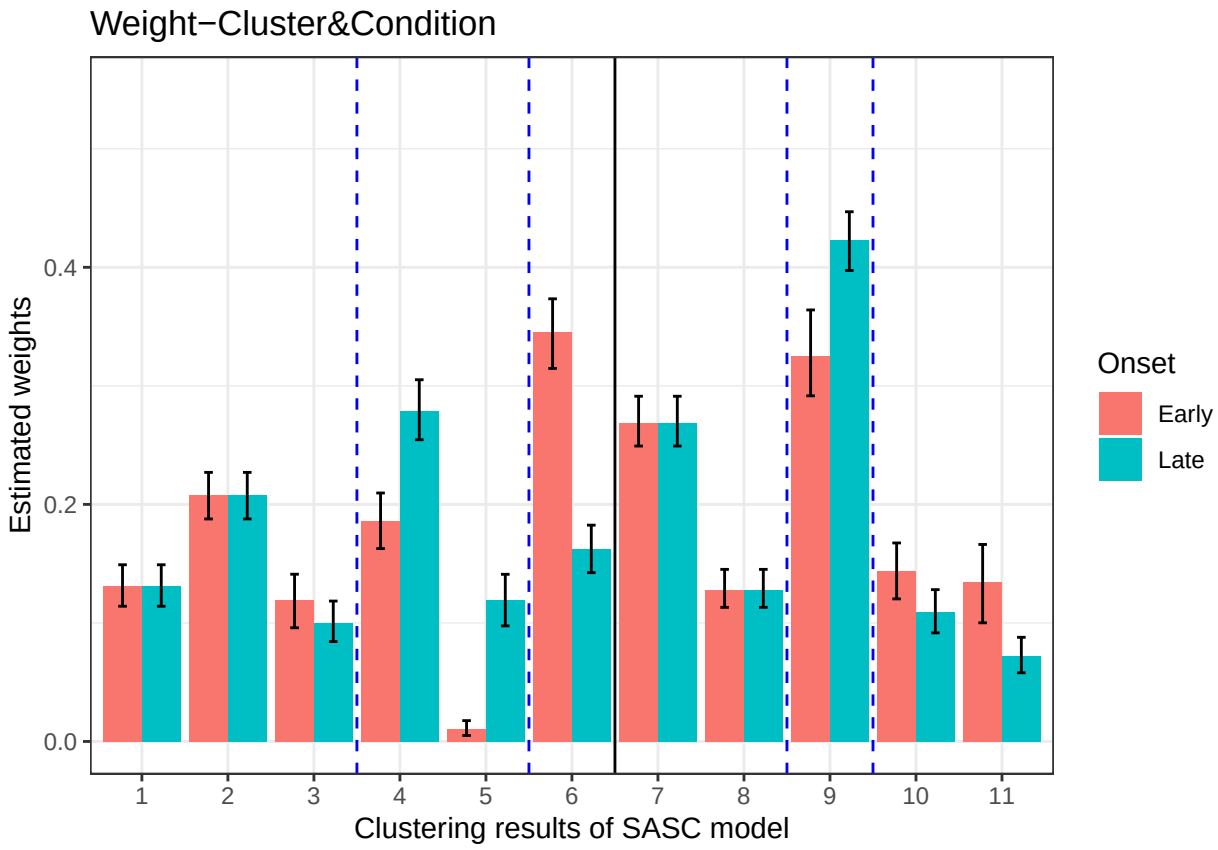
```
output <- input
output@meta.data$Cluster888 <- cluster

clusterinfo <- list(cluster_type_CLE = cluster_type_CLE,
                    xi_est = xi_est,
                    prob_in_h_mat = prob_in_h_mat,
                    weights = weights,
```

```
                    weights_mat = weights_mat)
saveRDS(output, file = paste0(output_data_dir, "/output888.rds"))
saveRDS(clusterinfo, file = paste0(output_data_dir, "/clusterinfo.rds"))
```

Visualization:

```
library(ggplot2); theme_set(theme_bw())
library(ggpubr)

# colors
pal <- c("#ffb6db", "#33A02C", "#b66dff", "#FEC44F", "#41B6C4", "#8E0152", "#0868AC", "#807DBA", "#E729
         "#00441B", "#525252", "#4D9221", "#8B5742", "#D8DAEB", "#7cdd2d", "#980043", "#8C96C6", "#EC70
         "#FDAE61", "#1D91C0", "#A6DBA0", "#4292C6", "#BF812D", "#01665E", "#41AB5D", "#FE9929", "#2525
names(pal) <- 1:30

pointsize <- 0.5

weights <- clusterinfo$weights
weights_mat <- clusterinfo$weights_mat
cluster_type_CLE <- clusterinfo$cluster_type_CLE
prob_in_h_mat <- clusterinfo$prob_in_h_mat
cluster <- output@meta.data$Cluster888
annotation <- output@meta.data$annotation_final
gene <- output@reductions$pca@cell.embeddings
gene_umap <- output@reductions$umap@cell.embeddings
onset <- output@meta.data$Onset
cell_names <- output@assays$RNA@counts@Dimnames[[2]]
# rename the groups to be L and E
onset_LE <- onset
onset_LE[which(onset == "LOCRC")] <- "L"
onset_LE[which(onset == "YOCRC")] <- "E"
n_clusters <- length(unique(cluster))
cluster_names <- as.character(1:n_clusters)


#########################
alpha <- 0.05
ci_low = c(apply(weights_mat[[1]], 2, quantile, alpha/2),
           apply(weights_mat[[2]], 2, quantile, alpha/2))

ci_high = c(apply(weights_mat[[1]], 2, quantile, (1 - alpha/2)),
            apply(weights_mat[[2]], 2, quantile, (1 - alpha/2)))


## Uncertainty of the weights
# plot
df_bar <- data.frame(Condition = c(rep("L" , n_clusters), rep("E" , n_clusters) ),
                     Onset = c(rep("Late" , n_clusters), rep("Early" , n_clusters) ),
                     Cluster = rep(as.character(1:(n_clusters)) , 2),
                     Weight = c(weights[1, ], weights[2, ]),
                     ci_low = ci_low,
                     ci_high = ci_high)
```

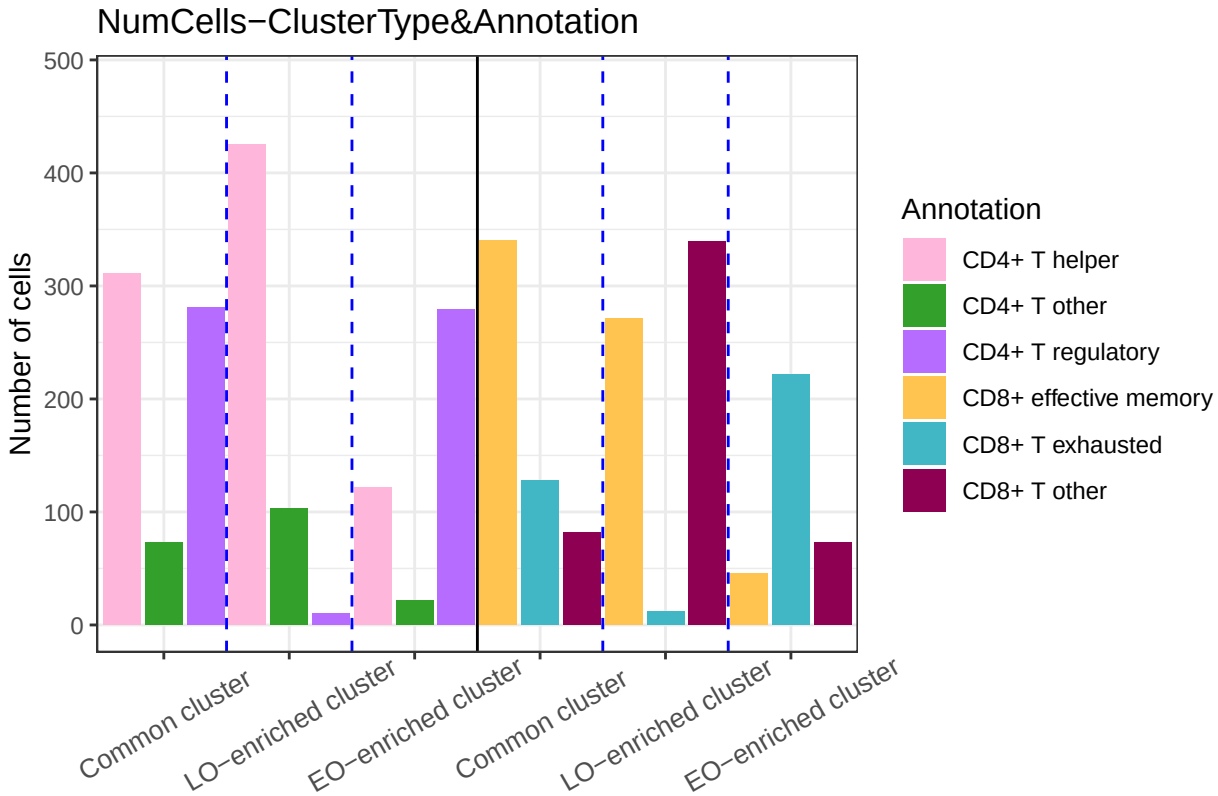
```r
# mean and error bar
p1 <- ggplot(df_bar, aes(fill=Onset, y=Weight, x=Cluster)) +
  geom_bar(position="dodge", stat="identity") +
  scale_x_discrete(limits = cluster_names) +
  geom_errorbar(aes(ymin=ci_low, ymax=ci_high),
                width=.2,
                position=position_dodge(.9)) +
  geom_vline(xintercept = 6.5,
             color = "black", linewidth=0.5) +
  geom_vline(xintercept = 3.5, linetype="dashed",
             color = "blue", linewidth=0.5) +
  geom_vline(xintercept = 5.5, linetype="dashed",
             color = "blue", linewidth=0.5) +
  geom_vline(xintercept = 8.5, linetype="dashed",
             color = "blue", linewidth=0.5) +
  geom_vline(xintercept = 9.5, linetype="dashed",
             color = "blue", linewidth=0.5) +
  ylim(c(0,0.55)) +
  xlab("Clustering results of SASC model") +
  ylab("Estimated weights") +
  ggtitle("Weight-Cluster&Condition")
p1
```




```r
ggsave(paste0(output_figure_dir, "/weights.pdf"),height = 3.5, width = 4.3)


####
```

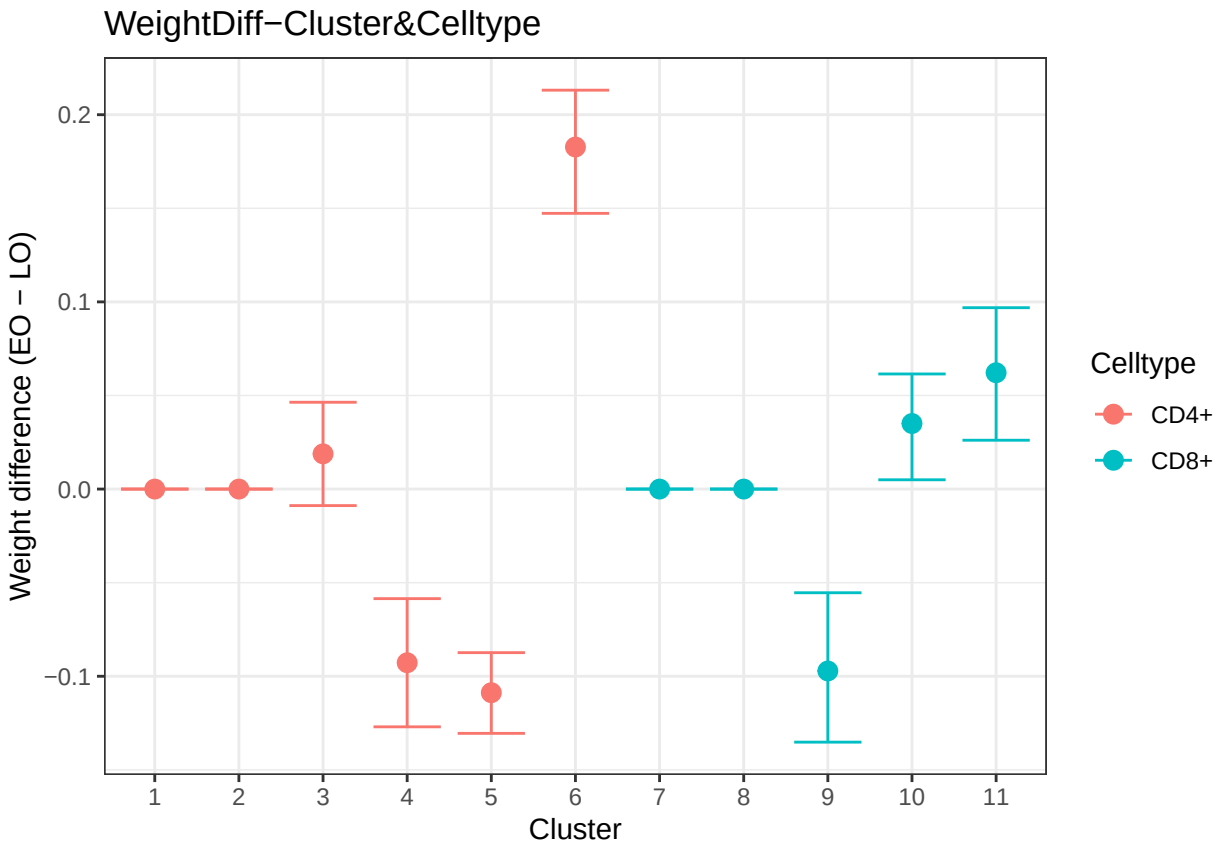
```r
## num of different celltypes in each cluster

celltypes <- annotation
types <- names(table(celltypes))
pal0 <- pal[1:length(types)]
names(pal0) <- types

celltypes_annot <- types
num_celltypes <- length(celltypes_annot)
num_celltypes_cluster <- c()
for (ii in 1:(n_clusters)) {
  for (jj in 1:num_celltypes) {
    num_celltypes_cluster <- c(num_celltypes_cluster, length(which(celltypes[cluster == ii] == celltypes
  }
}
cluster_temp <- c()
for (ii in 1:(n_clusters)) {
  cluster_temp <- c(cluster_temp, rep(ii, num_celltypes))
}

# plot
df_bar <- data.frame(Condition = rep(celltypes_annot , n_clusters),
                     Cluster = cluster_temp,
                     num_celltypes_cluster =num_celltypes_cluster)


####
## num of different celltypes in each clustertype

celltypes <- annotation
types <- names(table(celltypes))

celltypes_annot <- types
num_celltypes <- length(celltypes_annot)
num_celltypes_clustertype <- c()
clustertype <- rep(NA, length(cluster))
clustertype[which(cluster %in% which(cluster_type_CLE == "C"))] <- "C"
clustertype[which(cluster %in% which(cluster_type_CLE == "L"))] <- "L"
clustertype[which(cluster %in% which(cluster_type_CLE == "E"))] <- "E"
for (ii in unique(clustertype)) {
  for (jj in 1:num_celltypes) {
    num_celltypes_clustertype <- c(num_celltypes_clustertype, length(which(celltypes[clustertype == ii]
  }
}
cluster_temp <- c()
for (ii in unique(clustertype)) {
  cluster_temp <- c(cluster_temp, rep(ii, num_celltypes))
}

clustertype_names <- c("C_cd4T_hp", "C_cd4T_other", "C_cd4T_rg",
                       "L_cd4T_hp", "L_cd4T_other", "L_cd4T_rg",
                       "E_cd4T_hp", "E_cd4T_other", "E_cd4T_rg",
                       "C_CD8T_em", "C_CD8T_ex", "C_CD8T_other",
```

```
                      "L_CD8T_em", "L_CD8T_ex", "L_CD8T_other",
                      "E_CD8T_em", "E_CD8T_ex", "E_CD8T_other")

types_plot <- c("CD4+ T helper", "CD4+ T other", "CD4+ T regulatory",
                "CD8+ effective memory", "CD8+ T exhausted", "CD8+ T other")
pal0 <- pal[1:length(types_plot)]
names(pal0) <- types_plot
# plot
df_bar <- data.frame(Annotation = rep(types_plot, 3),
                     Cluster = paste0(cluster_temp, rep("_", length(cluster_temp)), rep(celltypes_annot
                     num_celltypes_clustertype =num_celltypes_clustertype)
df_bar$Cluster <- factor(df_bar$Cluster, levels = clustertype_names)
# Grouped bar plot
p3 <- ggplot(df_bar, aes(fill=Annotation, y=num_celltypes_clustertype, x=Cluster)) +
  geom_bar(stat="identity") +
  scale_fill_manual(values=pal0) +
  ylab("Number of cells") +
  theme(axis.text.x = element_text(angle = 30, vjust =0.6, hjust=0.5, size = 10)) +
  geom_vline(xintercept = 9.5,
             color = "black", linewidth=0.5) +
  geom_vline(xintercept = 3.5, linetype="dashed",
             color = "blue", linewidth=0.5) +
  geom_vline(xintercept = 6.5, linetype="dashed",
             color = "blue", linewidth=0.5) +
  geom_vline(xintercept = 12.5, linetype="dashed",
             color = "blue", linewidth=0.5) +
  geom_vline(xintercept = 15.5, linetype="dashed",
             color = "blue", linewidth=0.5) +
  ylim(c(0,480)) + xlab("") +
  scale_x_discrete(breaks=c("C_cd4T_other", "L_cd4T_other", "E_cd4T_other",
                            "C_CD8T_ex", "L_CD8T_ex", "E_CD8T_ex"),
                   labels=c("Common cluster", "LO-enriched cluster", "EO-enriched cluster",
                            "Common cluster", "LO-enriched cluster", "EO-enriched cluster")) +
  ggtitle("NumCells-ClusterType&Annotation")
p3
```



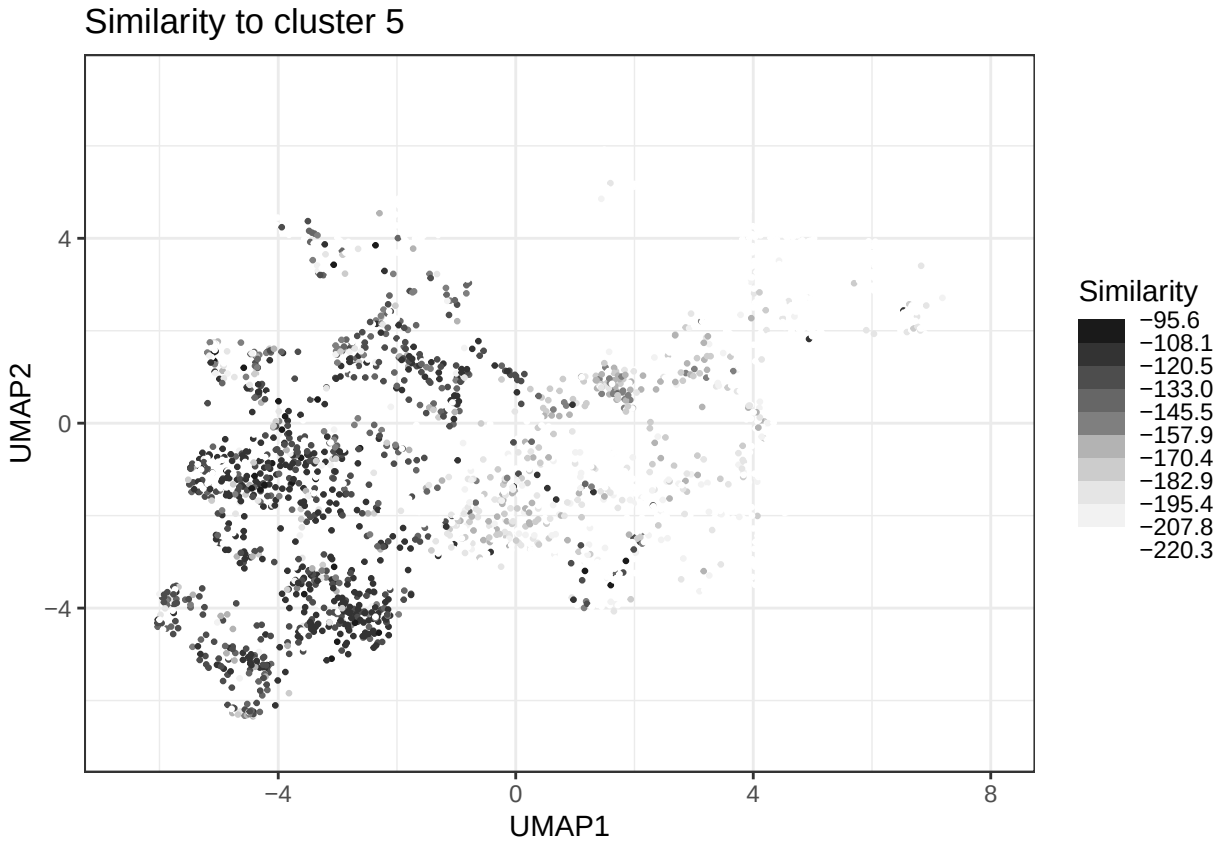

```
ggsave(paste0(output_figure_dir, "/num_cellstypes_clustertype.pdf"),height = 4, width = 5.5)


############## Annotate cluster types
num_celltypes_cluster_mat <- matrix(num_celltypes_cluster, ncol = num_celltypes, byrow = TRUE)
node_annot <- rep(NA, n_clusters)
cell_annot <- rep(NA, length(cell_names))
for (i in 1:n_clusters) {
  node_annot[i] <- types[which.max(num_celltypes_cluster_mat[i, ])]
  cell_annot[which(cluster == as.character(i))] <- node_annot[i]
}

cluster_annot <- rep(NA, n_clusters)
for (i in 1:n_clusters) {
  cluster_annot[i] <- node_annot[i]
}
names(cluster_annot) <- cluster_names <- as.character(1:(n_clusters))
# node_annot is the cluster annotations
# cell_annot is the cell annotations (the ones corresponding to the cluster assigned to)
df <- data.frame(Cluster = as.character(1:(n_clusters)),
                 weightLO_EO = -apply(weights_mat[[1]] - weights_mat[[2]], 2, mean),
                 ci_low = -apply(weights_mat[[1]] - weights_mat[[2]], 2, quantile, alpha/2),
                 ci_high = -apply(weights_mat[[1]] - weights_mat[[2]], 2, quantile, (1 - alpha/2)),
                 Celltype = c(rep("CD4+", 6), rep("CD8+", 5)))
p2 <- ggplot(df, aes(x=Cluster, y= weightLO_EO)) +
  geom_errorbar(width=0.8, aes(ymin=ci_low, ymax=ci_high, col = Celltype)) +
  # scale_color_manual(values=pal0) +
  geom_point(aes(col = Celltype), size = 3) +
```

```
  scale_x_discrete(limits = cluster_names) +
  ylab("Weight difference (EO - LO)") +
  ggtitle("WeightDiff-Cluster&Celltype")
p2
```



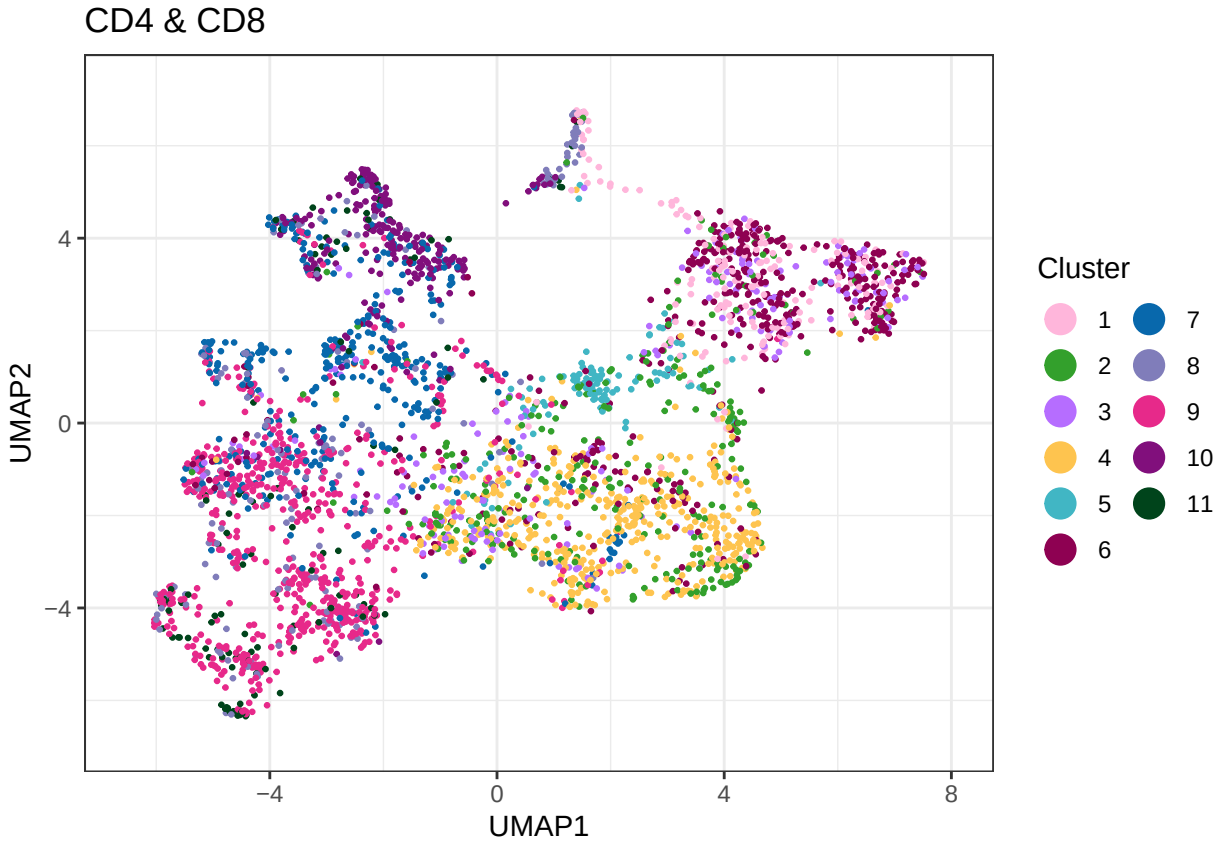

```
ggsave(paste0(output_figure_dir, "/weightsLO_EO_color.pdf"),height = 4, width = 5)


#######
### plot uncertainty
H <- 2

h_temp <- H + 3  # the cluster of interest
prob_in_h <- prob_in_h_mat[h_temp,]

df <- data.frame(PCA1 = gene[,1], PCA2 = gene[,2],
                 UMAP1 = gene_umap[,1], UMAP2 = gene_umap[,2],
                 Onset = onset_LE, cluster = cluster, Similarity = prob_in_h)

xrange <- c(min(df$UMAP1) - 0.5 , max(df$UMAP1) + 0.5)
yrange <- c(min(df$UMAP2) - 0.5 , max(df$UMAP2) + 0.5)

# df_naomit <- na.omit(df)
lim_lower <- round(as.numeric(quantile(na.omit(prob_in_h), 0.5)),1) - 0.1
lim_upper <- round(max(na.omit(prob_in_h)), 1) + 0.1
p_uncert <- ggplot(data=df)+
  geom_point(aes(UMAP1, UMAP2, colour = Similarity), size=pointsize) +
```

```
  binned_scale(aesthetics = "color",
               scale_name = "stepsn",
               palette = function(x) c("grey100", "grey95", "grey90", "grey80", "grey70", "grey50", "gr
               breaks = round(seq(lim_lower, lim_upper, length.out = 11), 1),
               limits = c(lim_lower, lim_upper),
               show.limits = TRUE,
               guide = "colorsteps") +
  ylim(yrange) + xlim(xrange) +
  ggtitle(paste0("Similarity to cluster ", h_temp))
p_uncert
```



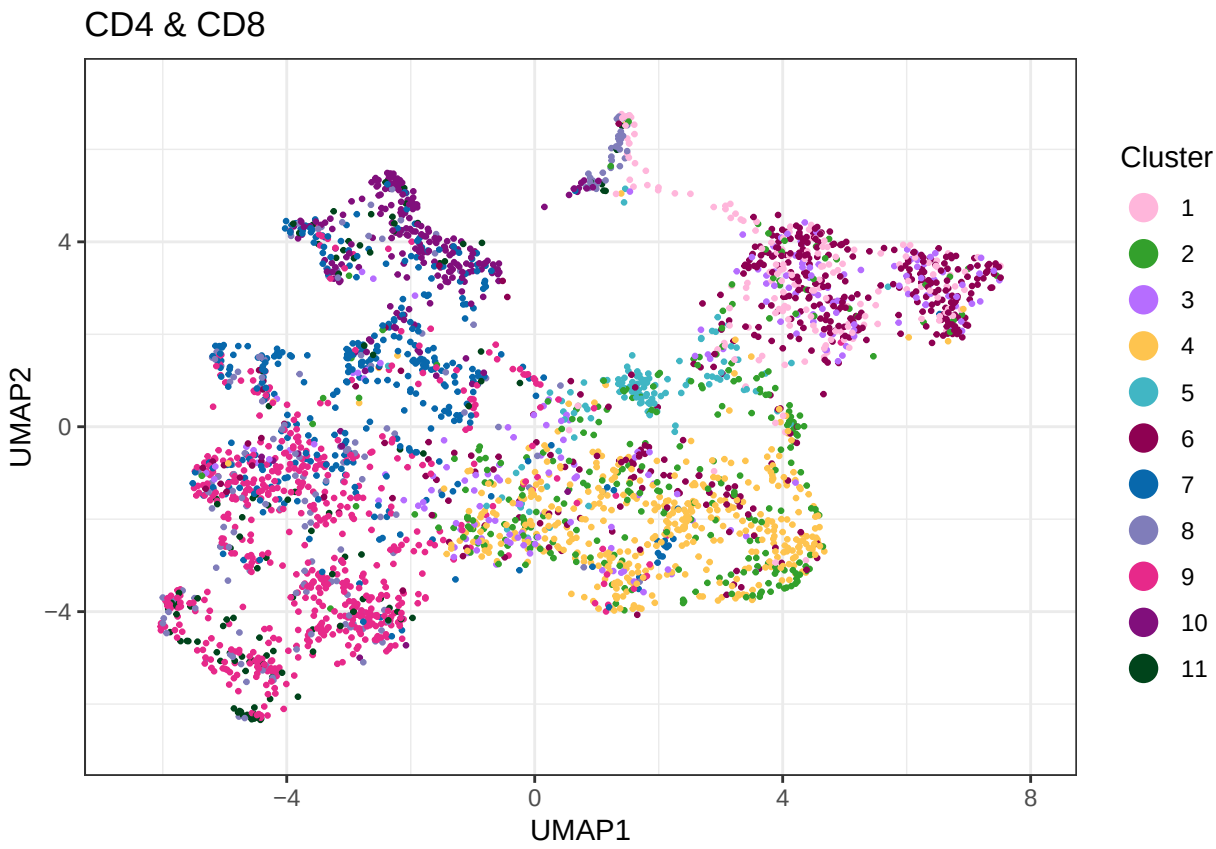

```
ggsave(paste0(output_figure_dir, "/Prob.pdf"),height = 4, width = 5)


#########
# plot in UMAP score

pointsize <- 0.5

# colors
pal <- pal[1:(n_clusters)]


##########
df <- data.frame(PCA1 = gene[,1], PCA2 = gene[,2],
                 UMAP1 = gene_umap[,1], UMAP2 = gene_umap[,2],
                 Onset = onset_LE, Cluster = cluster)
```

```
df$Cluster <- factor(df$Cluster, levels = as.character(1:n_clusters))

df_naomit <- na.omit(df)

p1_cluster <- ggplot(df_naomit, aes(x = UMAP1, y = UMAP2, color = Cluster)) +
  geom_point(size = pointsize) +
  guides(colour=guide_legend(override.aes=list(size = 5), ncol = 2)) +
  scale_color_manual(values=c(pal)) +
  ylim(yrange) + xlim(xrange) +
  ggtitle("CD4 & CD8")
p1_cluster
```

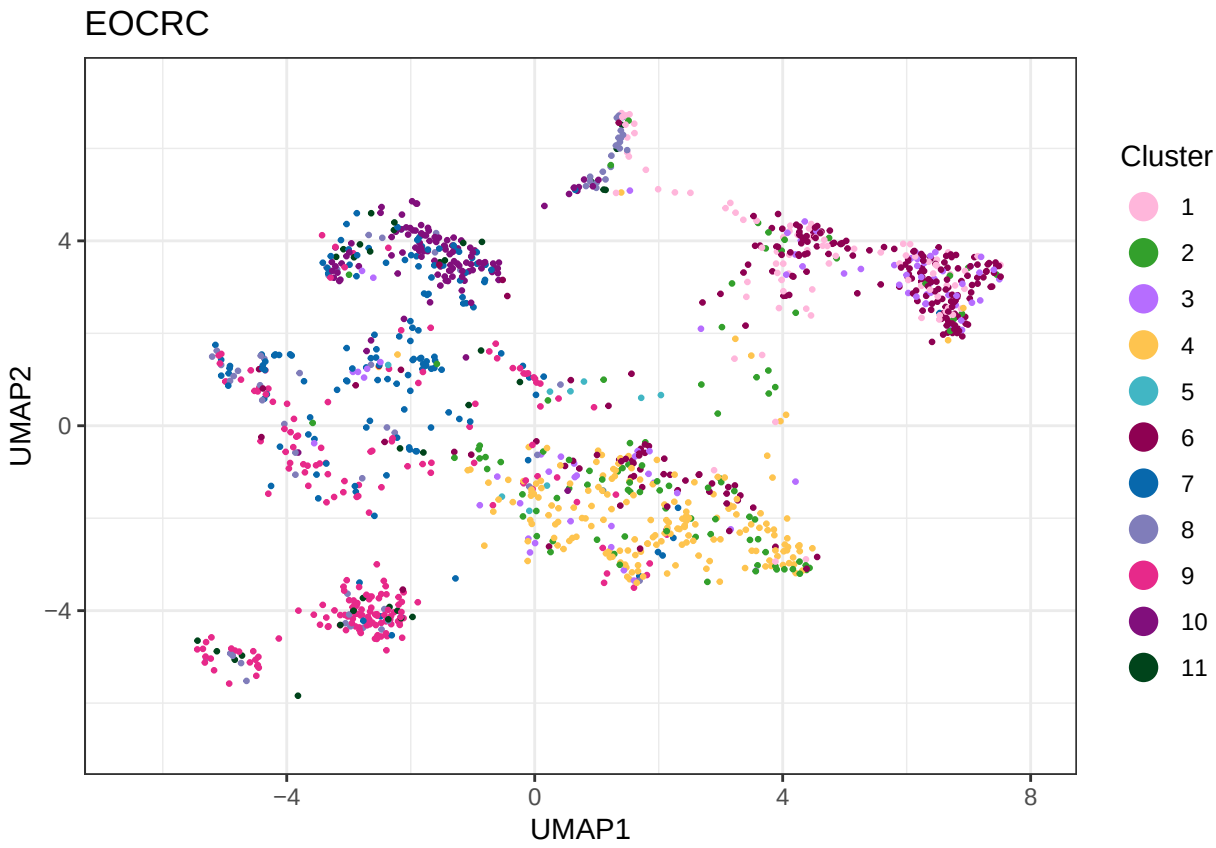



```
################################################
# UMAP plots

df <- data.frame(PCA1 = gene[,1], PCA2 = gene[,2],
                 UMAP1 = gene_umap[,1], UMAP2 = gene_umap[,2],
                 Onset = onset_LE, Cluster = cluster)
df$Cluster <- factor(df$Cluster, levels = as.character(1:n_clusters))
df_naomit <- na.omit(df)

p1 <- ggplot(df_naomit, aes(x = UMAP1, y = UMAP2, color = Cluster)) +
  geom_point(size = pointsize) +
  guides(colour=guide_legend(override.aes=list(size = 4.5))) +
  scale_color_manual(values=c(pal)) +
  ylim(yrange) + xlim(xrange) +
  ggtitle("CD4 & CD8")
```

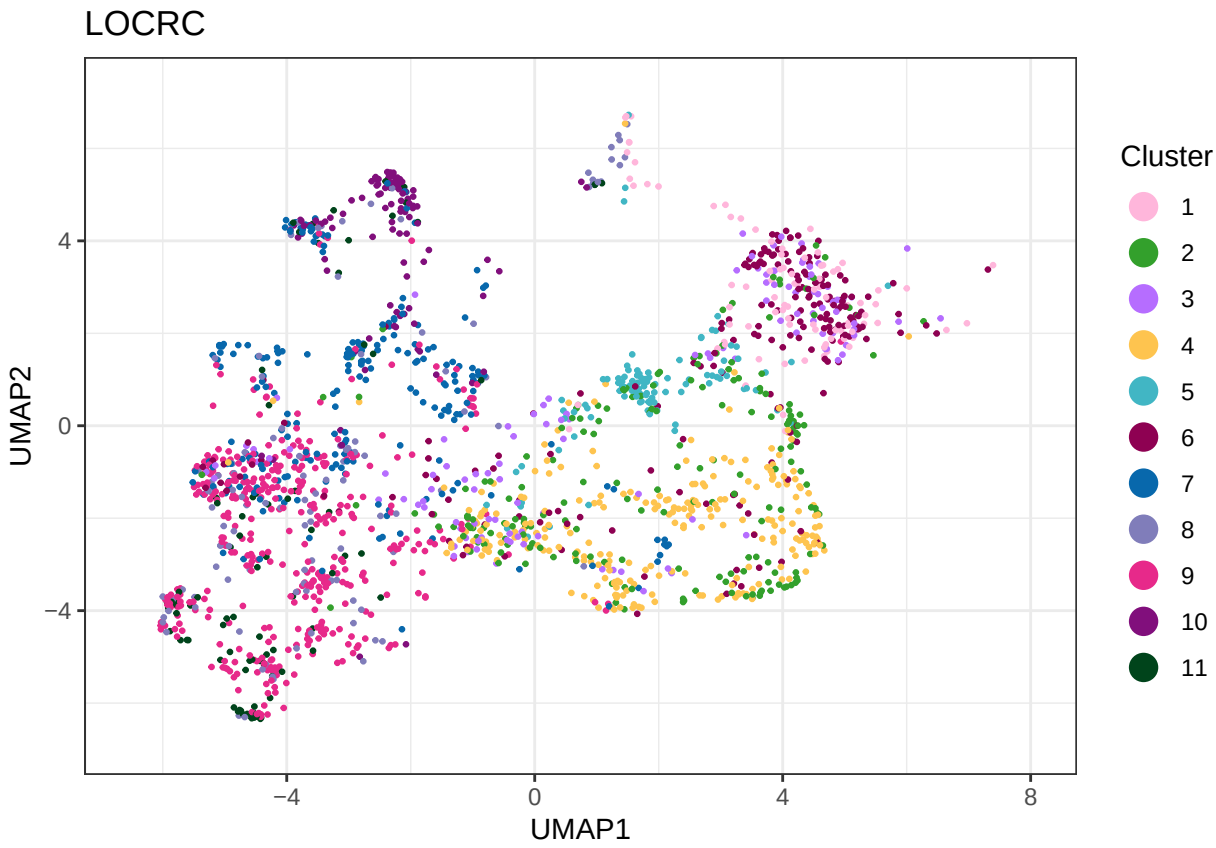
```
p1
```



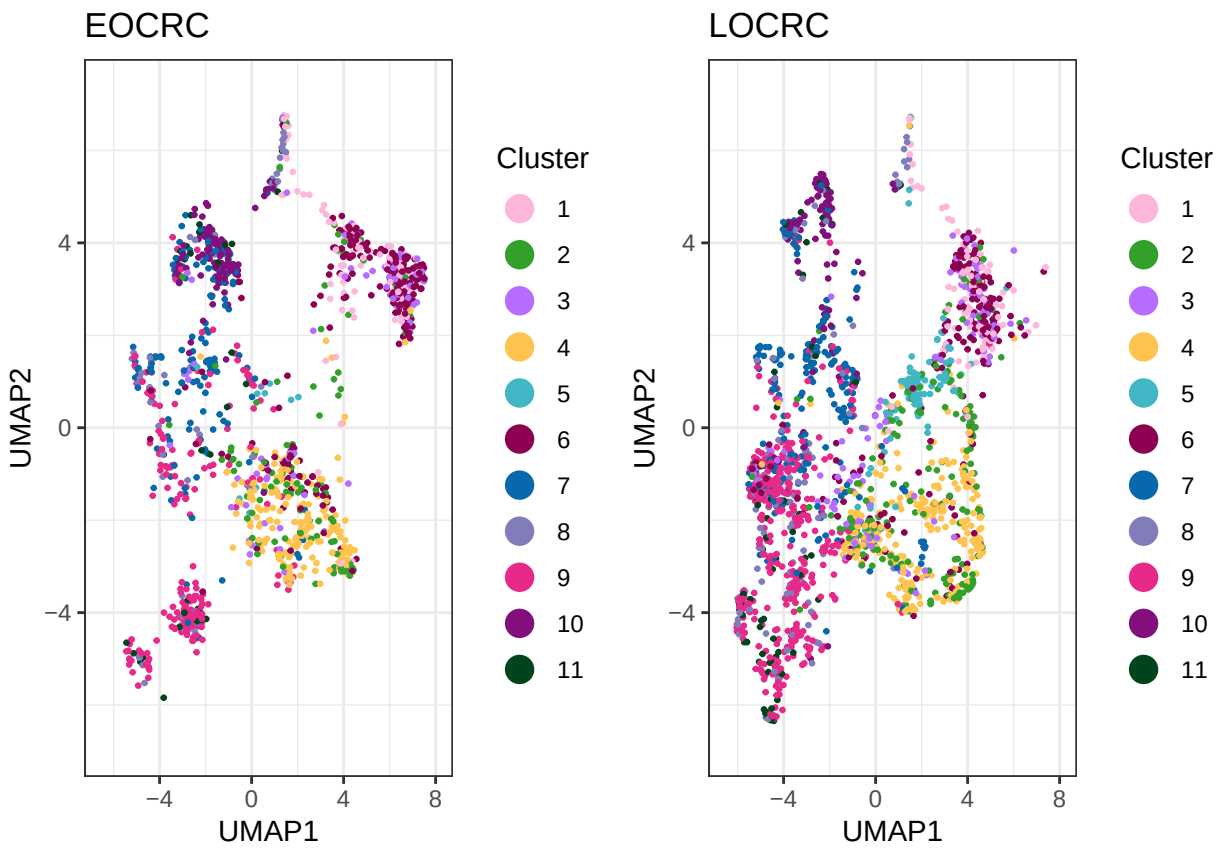

```
# plot cluster comparison

df_YOCRC <- df_naomit[which(df_naomit$Onset == "E"), ]
# plot
p2 <- ggplot(df_YOCRC, aes(x = UMAP1, y = UMAP2, color = Cluster)) +
  geom_point(size = pointsize) +
  guides(colour=guide_legend(override.aes=list(size = 4.5))) +
  scale_color_manual(values=c(pal)) +
  ylim(yrange) + xlim(xrange) +
  ggtitle("EOCRC")
p2
```




```
df_LOCRC <- df_naomit[which(df_naomit$Onset == "L"), ]
# plot
p3 <- ggplot(df_LOCRC, aes(x = UMAP1, y = UMAP2, color = Cluster)) +
  geom_point(size = pointsize) +
  guides(colour=guide_legend(override.aes=list(size = 4.5))) +
  scale_color_manual(values=c(pal)) +
  ylim(yrange) + xlim(xrange) +
  ggtitle("LOCRC")
p3
```




```
plot <- ggarrange(p2, p3, ncol = 2, nrow = 1, align = "hv")
plot
```




```
ggsave(filename = paste0(output_figure_dir, "/umap_all2.pdf"), width = 7.5, height = 3.5)

##################

YOcluster_ind <- which(cluster_type_CLE == "E")
## YOcluster
df_YOCRC_YOcluster <- df_naomit[which(df_naomit$Onset == "E" &
                                      as.numeric(df_naomit$Cluster) %in% YOcluster_ind ), ]
# plot
p1 <- ggplot(df_YOCRC_YOcluster, aes(x = UMAP1, y = UMAP2, color = Cluster)) +
  geom_point(size = pointsize) +
  guides(colour=guide_legend(override.aes=list(size = 5))) +
  scale_color_manual(values=c(pal[YOcluster_ind])) +
  ylim(yrange) + xlim(xrange) +
  ggtitle("EOCRC EOcluster")

df_LOCRC_YOcluster <- df_naomit[which(df_naomit$Onset == "L" &
                                      as.numeric(df_naomit$Cluster) %in% YOcluster_ind ), ]
# plot
p2 <- ggplot(df_LOCRC_YOcluster, aes(x = UMAP1, y = UMAP2, color = Cluster)) +
  geom_point(size = pointsize) +
  guides(colour=guide_legend(override.aes=list(size = 5))) +
  scale_color_manual(values=c(pal[YOcluster_ind])) +
  ylim(yrange) + xlim(xrange) +
  ggtitle("LOCRC EOcluster")

LOcluster_ind <- which(cluster_type_CLE == "L")
```

```r
## LOcluster
df_YOCRC_LOcluster <- df_naomit[which(df_naomit$Onset == "E" &
                                      as.numeric(df_naomit$Cluster) %in% LOcluster_ind), ]
# plot
p3 <- ggplot(df_YOCRC_LOcluster, aes(x = UMAP1, y = UMAP2, color = Cluster)) +
  geom_point(size = pointsize) +
  guides(colour=guide_legend(override.aes=list(size = 5))) +
  scale_color_manual(values=c(pal[LOcluster_ind])) +
  ylim(yrange) + xlim(xrange) +
  ggtitle("EOCRC LOcluster")

df_LOCRC_LOcluster <- df_naomit[which(df_naomit$Onset == "L" &
                                      as.numeric(df_naomit$Cluster) %in% LOcluster_ind), ]
# plot
p4 <- ggplot(df_LOCRC_LOcluster, aes(x = UMAP1, y = UMAP2, color = Cluster)) +
  geom_point(size = pointsize) +
  guides(colour=guide_legend(override.aes=list(size = 5))) +
  scale_color_manual(values=c(pal[LOcluster_ind])) +
  ylim(yrange) + xlim(xrange) +
  ggtitle("LOCRC LOcluster")

## COMMONcluster
COcluster_ind <- which(cluster_type_CLE == "C")

df_YOCRC_COMMONcluster <- df_naomit[which(df_naomit$Onset == "E" &
                                        as.numeric(df_naomit$Cluster) %in% COcluster_ind), ]
# plot
p5 <- ggplot(df_YOCRC_COMMONcluster, aes(x = UMAP1, y = UMAP2, color = Cluster)) +
  geom_point(size = pointsize) +
  guides(colour=guide_legend(override.aes=list(size = 5))) +
  scale_color_manual(values=c(pal[COcluster_ind])) +
  ylim(yrange) + xlim(xrange) +
  ggtitle("EOCRC COMMONcluster")

df_LOCRC_COMMONcluster <- df_naomit[which(df_naomit$Onset == "L" &
                                        as.numeric(df_naomit$Cluster) %in% COcluster_ind), ]
# plot
p6 <- ggplot(df_LOCRC_COMMONcluster, aes(x = UMAP1, y = UMAP2, color = Cluster)) +
  geom_point(size = pointsize) +
  guides(colour=guide_legend(override.aes=list(size = 5))) +
  scale_color_manual(values=c(pal[COcluster_ind])) +
  ylim(yrange) + xlim(xrange) +
  ggtitle("LOCRC COMMONcluster")

plot <- ggarrange(p5, p6, p3, p4, p1, p2, ncol = 2, nrow = 3, align = "v")
ggsave(filename = paste0(output_figure_dir, "/umap_clusters.pdf"), width = 7.5, height = 9)
```