scRNA-seq Analysis of Cardiac Organoid

10-27-2023

Project description

The cells here are from cardiac organoids made by mixing 15% endocardial cells and 85% cardiomyocytes. These organoids are then treated with factors to induce trabeculation. In our analysis so far, we have noticed that all group except for LVNC respond to this stimulation.

We are interested to see what the differences between control and cardiomyopathy group are. Secondly, within the cardiomyopathy group we want to see what the drivers of LVNC are. In another word, what are the differences between LVNC and other cardiomyopathies (DCM and HCM).

Here are the sample grouping:

- Control: S19-3, S25-3, S26-3
- Cardiomyopathy (DCM): GW10, GW53, GW168
- Cardiomyopathy (HCM): GW129, GW167, GW169
- Cardiomyopathy (LVNC): GW30, GW64, GW159

The first question is to identify differences between the control and cardiomyopathy groups. The second and more significant question pertains to the differences between LVNC and the other cardiomyopathy groups. Specifically, we are interested in the trabecular/compaction process and how trabeculations differ between LVNC and the two other cardiomyopathy groups.

The raw reads from two sub-libraries were processed with Parse Biosciences spilt-pipe v1.0.6p using GRCh38 reference genome and with default parameters.

Analysis for Q1: The first question is to identify differences between the control and cardiomyopathy groups.

Loading libraries and setting location paths

```
# remove.packages('Matrix')
## install.packages("Matrix", repos="http://R-Forge.R-project.org")
# install.packages("Matrix")
library("Matrix")
library(Seurat)

## Attaching SeuratObject
```

```
# packageVersion("Seurat")
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(ggplot2)
library(cowplot)
library(readr)
# Sys.setenv(PKG_CONFIG_PATH="/fftw/install/3.3.10/lib/pkgconfig")
# install.packages("metap", dependencies = TRUE,)
# library("qqconf")
# library("metap")
options(future.globals.maxSize = 8000 * 1024^2)

## On HPCF
# hpcf_interactive -n 1 -R "rusage[mem=100000]" -q rhel8_interactive
# cd /RNAseq/common/scRNAseq_Paul_cardiomyopathy_round2
# module load fftw/3.3.10


# setwd("/analysis/combined_figures/")
setwd("/RNAseq/common/scRNAseq_Paul_cardiomyopathy_round2/newvolume/analysis/
combined_figures/")

# rm(list = ls())

# fig_path <- '/analysis/combined_figures/'
fig_path <-
"/RNAseq/common/scRNAseq_Paul_cardiomyopathy_round2/newvolume/analysis/combin
ed_figures/"

# data_path <- '/analysis/combined_figures/'
data_path <-
"/RNAseq/common/scRNAseq_Paul_cardiomyopathy_round2/newvolume/analysis/combin
ed_figures/"
```

Below we've included a few convenience functions for saving images and reading/writing Seurat object to disk. When we're working with larger datasets, it's usually a good idea to save progress after computationally intensive steps so we can back track if we wish to do so.

Reading in data

After reading in the data we'll perform basic filtering on our expression matrix to remove low quality cells and uninformative genes. The parameter "min_genes" will keep cells that have at least 100 genes, and similarly, "min_cells" will keep genes that are expressed in at least 2 cells. Note: Seurat version 4.1 includes a convenience function to read Parse data from the DGE folder. If we would like to use this function, please skip the code block below and see the section "Reading in data with Seurat >= 4.1"

```
DGE_folder <-
"/RNAseq/common/scRNAseq_Paul_cardiomyopathy_round2/newvolume/analysis/S_comb
ined/all-well/DGE_filtered/"

# split-pipe versions older than 1.1.0 used "DGE.mtx"
mat <- readMM(paste0(DGE_folder, "DGE.mtx"))

cell_meta <- read.delim(paste0(DGE_folder, "cell_metadata.csv"),
                  stringsAsFactor = FALSE, sep = ",")
genes <- read.delim(paste0(DGE_folder, "all_genes.csv"),
                    stringsAsFactor = FALSE, sep = ",")

cell_meta$bc_wells <- make.unique(cell_meta$bc_wells, sep = "_dup")
rownames(cell_meta) <- cell_meta$bc_wells
genes$gene_name <- make.unique(genes$gene_name, sep = "_dup")

# Setting column and rownames to expression matrix
colnames(mat) <- genes$gene_name
rownames(mat) <- rownames(cell_meta)
mat_t <- t(mat)

# Remove empty rownames, if they exist
mat_t <- mat_t[(rownames(mat_t) != ""),]

# Seurat version 5 or greater uses "min.features" instead of "min.genes"
pbmc <- CreateSeuratObject(mat_t, min.features = 100, min.cells = 2,
meta.data = cell_meta)
```

When we create our Seurat object with the plate well numbers (column names in the expression matrix) from the experiment will automatically be assigned to the cell identity slot. In other words, the program assumes this how we want to initially classify our cells. In general, we would like to avoid this behavior so there isn't a bias towards a particular cell class when removing outliers.

```
# Setting our initial cell class to a single type, this will changer after
clustering.
pbmc@meta.data$orig.ident <- factor(rep("pbmc", nrow(pbmc@meta.data)))
Idents(pbmc) <- pbmc@meta.data$orig.ident
#
```

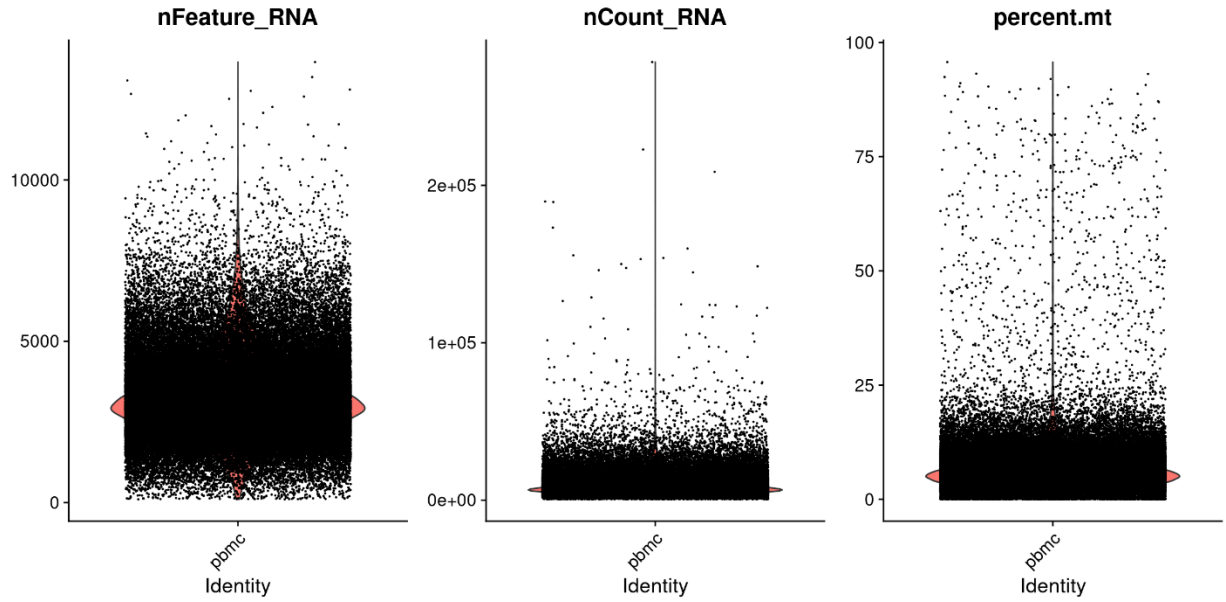
```
# # SaveObject(pbmc, "seurat_obj_before_QC_CMP_vs_control")
# pbmc <- ReadObject("seurat_obj_before_QC_CMP_vs_control")
```

Cell quality control

In this step we'll perform cell quality control to prevent outlier cells from influencing downstream analyses. Cells that have unusually high transcript or gene counts are very likely to be multiplets, which is a term for two or more cells that have identical barcodes. Conversely, cells that have very low transcript or genes counts are a consequence of barcoding cells with damage membranes, or barcoding ambient RNA.

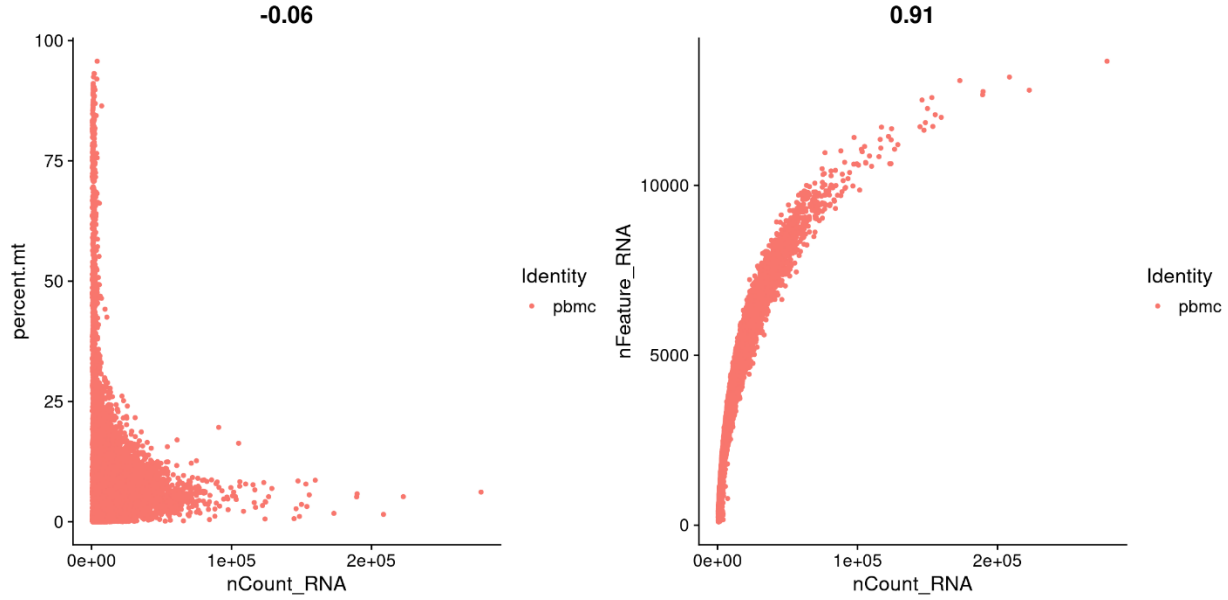
Filtering outliers can be accomplished by generating a violin plot for each cell feature and manually selecting the threshold we wish to remove cells. We'll also add another important cell feature that shows the percentage of mitochondrial genes expressed in each cell. Cells with high mitochondrial gene percentages should be removed, as they are likely to have lost cytoplasmic RNA from lysis or may have increased apoptosis (Luecken and Theis 2019)

```
pbmc[["percent.mt"]] <- PercentageFeatureSet(pbmc, pattern = "^MT-")
plot <- VlnPlot(pbmc, pt.size = 0.10,
features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)
plot
# SaveFigure(plot, "vln_QC_CMP_vs_control", width = 12, height = 6)
```




```
plot1 <- FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 =
"percent.mt")
plot1
plot2 <- FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 =
```

```
"nFeature_RNA")
plot2

# plot_grid(plot1, plot2)
# SaveFigure((plot1 + plot2),"scatter_QC_CMP_vs_control", width = 12, height
= 6, res = 300)
```



Let's break down the conditions in the subset argument below:

nFeature_RNA is the number of genes detected in each cell. nCount_RNA is the total number of molecules detected within a cell. Low nFeature_RNA for a cell indicates that it may be dead/dying or an empty droplet. High nCount_RNA and/or nFeature_RNA indicates that the "cell" may in fact be a doublet (or multiplet). In combination with % of mitochondrial reads, removing outliers from these groups removes most doublets/dead cells/empty droplets, hence why filtering is a common pre-processing step.

```
# Perform the filtering
# pbmc <- subset(pbmc, subset = nFeature_RNA < 5000 & nCount_RNA < 20000 &
percent.mt < 15)
```

QC 1. Normalizing the data

After removing unwanted cells from the dataset, the next step is to normalize the data. By default, we employ a global-scaling normalization method LogNormalize that normalizes the feature expression measurements for each cell by the total expression, multiplies this by a scale factor (10,000 by default), and log-transforms the result. Normalized values are stored in pbmc[["RNA"]]@data.

QC 2. Identification of highly variable features

We next calculate a subset of features that exhibit high cell-to-cell variation in the dataset (i.e, they are highly expressed in some cells, and lowly expressed in others). We and others have found that focusing on these genes in downstream analysis helps to highlight biological signal in single-cell datasets. Our procedure in Seurat3 is described in detail here, and improves on previous versions by directly modeling the mean-variance relationship inherent in single-cell data, and is implemented in the FindVariableFeatures function. By default, we return 2,000 features per dataset. These will be used in downstream analysis, like PCA.

Each dataset is independently normalized by restricting cells with:

a. Less than 5,000 expressed genes (ensures the exclusion of low-quality cells or empty droplets from further analysis)

b. Fewer than 20,000 total gene counts (helps remove potential multiplets or cells with damaged membranes, reducing the risk of introducing)

c. Mitochondrial gene percentages under 15%. A high percentage of mitochondrial genes in a cell may indicate compromised cell health or increased apoptosis

This process involves normalizing the data using a log transformation with a scaling factor of 10,000 and selecting 2,000 variable features for further analysis.

```
# split the dataset into a list of seurat objects by samples
ifnb.list <- SplitObject(pbmc, split.by = "sample")

sample_name <- names(ifnb.list)
## For question 1
for (sample_name in names(ifnb.list)) {
  seurat_obj <- ifnb.list[[sample_name]]

  # Assuming you have a condition column in the Seurat object that indicates
"Case" or "Control"
  # Replace "condition_column_name" with the actual name of your condition
column
  # Assuming the condition is defined as "control" if the sample name
contains "S19-3|S25-3|S26-3"
  seurat_obj$grouping_var1 <- ifelse(grepl("S19-3|S25-3|S26-3", sample_name),
"control", "cardiomyopathy")
  Idents(seurat_obj) <-  ifelse(grepl("S19-3|S25-3|S26-3", sample_name),
"control", "cardiomyopathy")
  ifnb.list[[sample_name]] <- seurat_obj
}
```

```
# normalize and identify variable features for each dataset independently
ifnb.list <- lapply(X = ifnb.list, FUN = function(x) {
    x <- subset(x, subset = nFeature_RNA < 5000 & nCount_RNA < 20000 &
percent.mt < 15)
    x <- NormalizeData(x, normalization.method = "LogNormalize", scale.factor
= 10000)
    x <- FindVariableFeatures(x, selection.method = "vst", nfeatures = 2000)
})


# select features that are repeatedly variable across datasets for
integration
features <- SelectIntegrationFeatures(object.list = ifnb.list)
```

Perform integration

We then identify anchors using the FindIntegrationAnchors() function, which takes a list of Seurat objects as input, and use these anchors to integrate the two datasets together with IntegrateData().

```
immune.anchors <- FindIntegrationAnchors(object.list = ifnb.list,
anchor.features = features)
# this command creates an 'integrated' data assay
immune.combined <- IntegrateData(anchorset = immune.anchors)

# SaveObject(immune.combined, "seurat_obj_post_integration_CMP_vs_control")
# immune.combined <- ReadObject("seurat_obj_post_integration_CMP_vs_control")
```

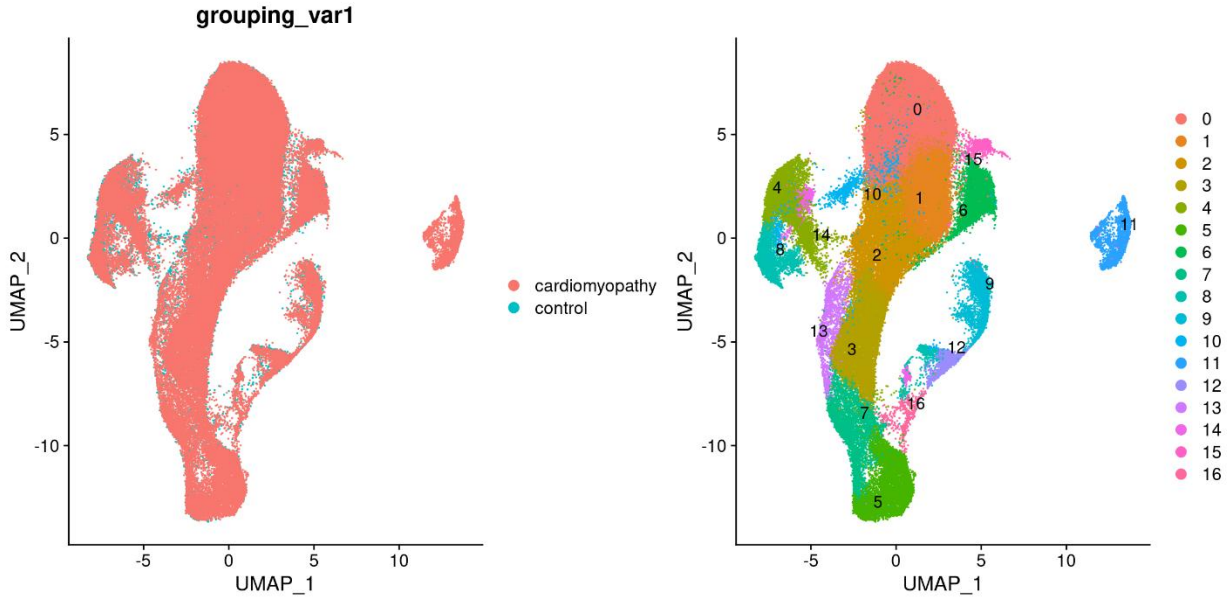
Perform an integrated analysis

Now we can run a single integrated analysis on all cells!

specify that we will perform downstream analysis on the corrected data. Note that the original unmodified data still resides in the 'RNA' assay.

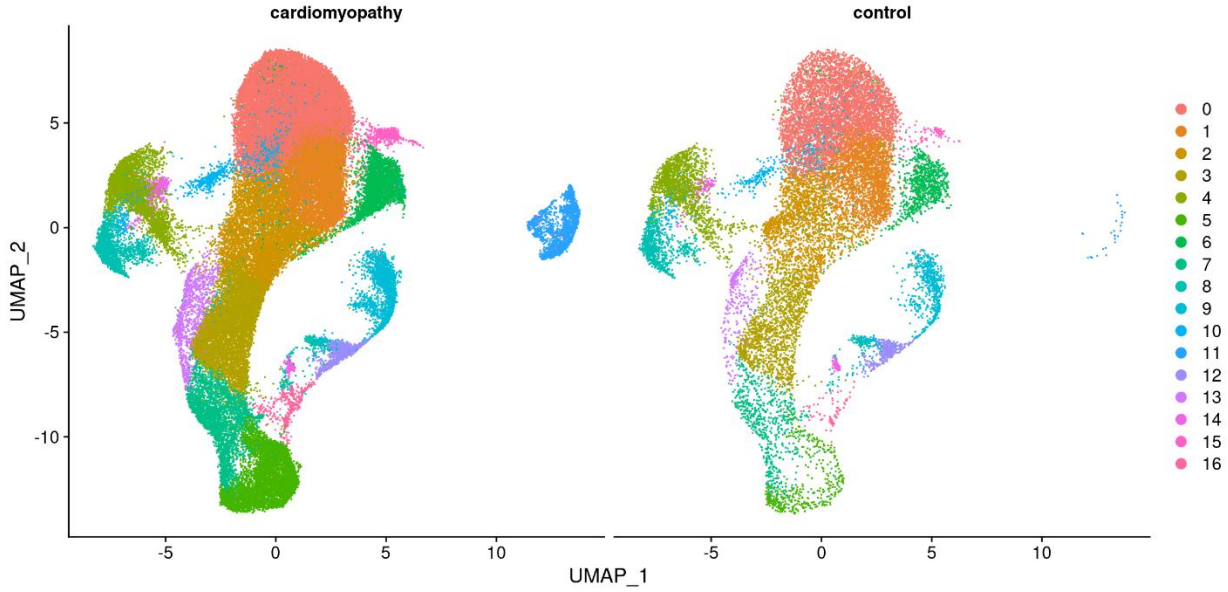
```
DefaultAssay(immune.combined) <- "integrated"

# Run the standard workflow for visualization and clustering
immune.combined <- ScaleData(immune.combined, verbose = FALSE)
immune.combined <- RunPCA(immune.combined, npcs = 30, verbose = FALSE)
immune.combined <- RunUMAP(immune.combined, reduction = "pca", dims = 1:30)
immune.combined <- FindNeighbors(immune.combined, reduction = "pca", dims =
1:30)
immune.combined <- FindClusters(immune.combined, resolution = 0.5)
```

```
# SaveObject(immune.combined,
"seurat_obj_post_integration_postCluster_CMP_vs_control")
# immune.combined <-
ReadObject("seurat_obj_post_integration_postCluster_CMP_vs_control")

# Visualization
p1 <- DimPlot(immune.combined, reduction = "umap", group.by =
"grouping_var1")
p2 <- DimPlot(immune.combined, reduction = "umap", label = TRUE, repel =
TRUE)
# p1 + p2
SaveFigure(p1+p2,"dimplot_cluster_CMP_vs_control", width = 12, height = 6,
res = 300)
```



To visualize the two conditions side-by-side, we can use the split.by argument to show each condition colored by cluster.

```
p1 <- DimPlot(immune.combined, reduction = "umap", split.by =
"grouping_var1")
SaveFigure(p1,"dimplot_cluster_conditions_side_by_side_CMP_vs_control", width
= 12, height = 6, res = 300)
```



Identify conserved cell type markers

To identify canonical cell type marker genes that are conserved across conditions, we can use the FindConservedMarkers() function. This function performs differential gene expression testing for each dataset/group and combines the p-values using meta-analysis methods from the MetaDE R package

We can explore these marker genes for each cluster and use them to annotate our clusters as specific cell types.

```
# For performing differential expression after integration, we switch back to
the original data
DefaultAssay(immune.combined) <- "RNA"
# loop over clusters
top.markers <- {}
top.markers.df <- list()
identities <- sort(unique(Idents(immune.combined)))
for(i in 1:length(identities)){
ident <- identities[i]
print(paste0("Doing identity: ", ident))
immune.combined_markers <- FindConservedMarkers(immune.combined, ident.1 =
ident, grouping.var = "grouping_var1", verbose = FALSE)
top.markers.tmp <- rownames(head(immune.combined_markers,1))
top.markers <- c(top.markers, top.markers.tmp)
top.markers.df.tmp <- immune.combined_markers
top.markers.df.tmp$cluster <- ident
top.markers.df[[i]] <- top.markers.df.tmp
}
```

```
# SaveObject(top.markers, "seurat_obj_top_markers_CMP_vs_control")
# SaveObject(top.markers.df,
"seurat_obj_top_markers_dataframe_list_CMP_vs_control")
```

Printing top 4 genes from each cluster..

```
top.markers <- ReadObject("seurat_obj_top_markers_CMP_vs_control")
top.markers.df <-
ReadObject("seurat_obj_top_markers_dataframe_list_CMP_vs_control")

lapply(top.markers.df, head, 4)
```

```
## [[1]]
##         control_p_val control_avg_log2FC control_pct.1 control_pct.2
## GRAMD1B             0          0.7418410         0.793         0.551
## MYOM2               0          0.8965687         0.688         0.354
## PLEKHA5             0          0.8045169         0.988         0.905
## ATP2B1              0          0.9724600         0.870         0.669
##         control_p_val_adj cardiomyopathy_p_val cardiomyopathy_avg_log2FC
## GRAMD1B                 0                    0                 0.9735223
## MYOM2                   0                    0                 1.1002873
## PLEKHA5                 0                    0                 0.8875826
## ATP2B1                  0                    0                 1.1325842
##         cardiomyopathy_pct.1 cardiomyopathy_pct.2 cardiomyopathy_p_val_adj
## GRAMD1B                0.807                0.484                        0
## MYOM2                  0.639                0.218                        0
## PLEKHA5                0.988                0.868                        0
## ATP2B1                 0.891                0.636                        0
##         max_pval minimump_p_val cluster
## GRAMD1B        0              0       0
## MYOM2          0              0       0
## PLEKHA5        0              0       0
## ATP2B1         0              0       0
##
## [[2]]
##        control_p_val control_avg_log2FC control_pct.1 control_pct.2
## MYH7               0          0.6949304         0.998         0.869
## PALLD              0          0.5033292         1.000         0.987
## CYP2J2             0          0.7422025         0.468         0.147
## ANKRD1             0          1.4169053         0.930         0.580
##        control_p_val_adj cardiomyopathy_p_val cardiomyopathy_avg_log2FC
## MYH7                   0                    0                 0.8196450
## PALLD                  0                    0                 0.7274541
## CYP2J2                 0                    0                 0.8230345
## ANKRD1                 0                    0                 1.4322350
##        cardiomyopathy_pct.1 cardiomyopathy_pct.2 cardiomyopathy_p_val_adj
## MYH7                  0.997                0.798                        0
## PALLD                 1.000                0.957                        0
```

```
## CYP2J2               0.537               0.165                        0
## ANKRD1               0.939               0.566                        0
##        max_pval minimump_p_val cluster
## MYH7          0              0       1
## PALLD         0              0       1
## CYP2J2        0              0       1
## ANKRD1        0              0       1
##
## [[3]]
##         control_p_val control_avg_log2FC control_pct.1 control_pct.2
## NAV3     0.000000e+00          1.2492099         0.874         0.537
## CACNA1C 7.057152e-287          0.6323161         0.999         0.979
## ADGRL3  2.516196e-251          0.7276695         0.980         0.843
## SORCS3  3.715961e-247          1.0927961         0.722         0.392
##         control_p_val_adj cardiomyopathy_p_val cardiomyopathy_avg_log2FC
## NAV3         0.000000e+00                    0                  1.3108221
## CACNA1C     2.862734e-282                    0                  0.7166100
## ADGRL3      1.020695e-246                    0                  0.7890516
## SORCS3      1.507379e-242                    0                  1.1904145
##         cardiomyopathy_pct.1 cardiomyopathy_pct.2 cardiomyopathy_p_val_adj
## NAV3                   0.889                0.537                        0
## CACNA1C                0.999                0.929                        0
## ADGRL3                 0.979                0.768                        0
## SORCS3                 0.809                0.420                        0
##              max_pval minimump_p_val cluster
## NAV3     0.000000e+00              0       2
## CACNA1C 7.057152e-287              0       2
## ADGRL3  2.516196e-251              0       2
## SORCS3  3.715961e-247              0       2
##
## [[4]]
##        control_p_val control_avg_log2FC control_pct.1 control_pct.2
## THSD7A             0          1.8221770         0.798         0.353
## VCAN               0         -2.1221459         0.502         0.949
## LTBP1              0          1.5689822         0.998         0.991
## CHRDL2             0          0.4001481         0.145         0.006
##        control_p_val_adj cardiomyopathy_p_val cardiomyopathy_avg_log2FC
## THSD7A                 0                    0                  1.5168109
## VCAN                   0                    0                 -2.4548483
## LTBP1                  0                    0                  1.7126566
## CHRDL2                 0                    0                  0.5135333
##        cardiomyopathy_pct.1 cardiomyopathy_pct.2 cardiomyopathy_p_val_adj
## THSD7A                0.799                0.390                        0
## VCAN                  0.525                0.919                        0
## LTBP1                 0.996                0.953                        0
## CHRDL2                0.180                0.010                        0
##        max_pval minimump_p_val cluster
## THSD7A        0              0       3
## VCAN          0              0       3
## LTBP1         0              0       3
```

```
## CHRDL2        0              0       3
##
## [[5]]
##       control_p_val control_avg_log2FC control_pct.1 control_pct.2
## BRCA1             0          0.7373738         0.620         0.204
## POLA2             0          0.5699580         0.395         0.067
## POLQ              0          0.8419355         0.567         0.113
## MCM10             0          0.6843958         0.417         0.038
##       control_p_val_adj cardiomyopathy_p_val cardiomyopathy_avg_log2FC
## BRCA1                 0                    0                 0.7890302
## POLA2                 0                    0                 0.5692590
## POLQ                  0                    0                 0.9151750
## MCM10                 0                    0                 0.7237640
##       cardiomyopathy_pct.1 cardiomyopathy_pct.2 cardiomyopathy_p_val_adj
## BRCA1                0.651                0.199                         0
## POLA2                0.425                0.062                         0
## POLQ                 0.593                0.088                         0
## MCM10                0.461                0.030                         0
##       max_pval minimump_p_val cluster
## BRCA1        0              0       4
## POLA2        0              0       4
## POLQ         0              0       4
## MCM10        0              0       4
##
## [[6]]
##        control_p_val control_avg_log2FC control_pct.1 control_pct.2
## CFH                0          0.8418136         0.270         0.004
## CD9                0          0.9396713         0.394         0.015
## COL9A2             0          0.8832300         0.292         0.006
## LAMC3              0          0.6594713         0.253         0.008
##        control_p_val_adj cardiomyopathy_p_val cardiomyopathy_avg_log2FC
## CFH                    0                    0                 1.9455299
## CD9                    0                    0                 0.3325983
## COL9A2                 0                    0                 0.5917556
## LAMC3                  0                    0                 0.3385183
##        cardiomyopathy_pct.1 cardiomyopathy_pct.2 cardiomyopathy_p_val_adj
## CFH                   0.623                0.012                         0
## CD9                   0.221                0.046                         0
## COL9A2                0.193                0.007                         0
## LAMC3                 0.115                0.009                         0
##        max_pval minimump_p_val cluster
## CFH           0              0       5
## CD9           0              0       5
## COL9A2        0              0       5
## LAMC3         0              0       5
##
## [[7]]
##        control_p_val control_avg_log2FC control_pct.1 control_pct.2
## TPM1               0           1.016163         0.997         0.993
## ACTC1              0           1.780882         0.986         0.887
```

```
## HSPB7              0           1.737618         0.956         0.793
## MT-CO1             0           1.720846         0.990         0.958
##        control_p_val_adj cardiomyopathy_p_val cardiomyopathy_avg_log2FC
## TPM1                   0                    0                  1.214641
## ACTC1                  0                    0                  2.010189
## HSPB7                  0                    0                  1.916764
## MT-CO1                 0                    0                  1.699231
##        cardiomyopathy_pct.1 cardiomyopathy_pct.2 cardiomyopathy_p_val_adj
## TPM1                  0.994                0.961                        0
## ACTC1                 0.980                0.794                        0
## HSPB7                 0.954                0.708                        0
## MT-CO1                0.985                0.945                        0
##        max_pval minimump_p_val cluster
## TPM1          0              0       6
## ACTC1         0              0       6
## HSPB7         0              0       6
## MT-CO1        0              0       6
##
## [[8]]
##       control_p_val control_avg_log2FC control_pct.1 control_pct.2
## CDH19             0           1.667163         0.305         0.021
## MAP2              0           1.988847         0.778         0.177
## NEBL              0          -2.482120         0.505         0.969
## MMP2              0           1.363339         0.673         0.111
##       control_p_val_adj cardiomyopathy_p_val cardiomyopathy_avg_log2FC
## CDH19                 0                    0                 0.9558272
## MAP2                  0                    0                 1.8653163
## NEBL                  0                    0                -2.5994234
## MMP2                  0                    0                 1.1061560
##       cardiomyopathy_pct.1 cardiomyopathy_pct.2 cardiomyopathy_p_val_adj
## CDH19                0.171                0.018                        0
## MAP2                 0.716                0.180                        0
## NEBL                 0.469                0.904                        0
## MMP2                 0.674                0.171                        0
##       max_pval minimump_p_val cluster
## CDH19        0              0       7
## MAP2         0              0       7
## NEBL         0              0       7
## MMP2         0              0       7
##
## [[9]]
##        control_p_val control_avg_log2FC control_pct.1 control_pct.2
## NCAPD2             0          1.0612328         0.793         0.175
## ANLN               0          2.0312426         0.993         0.234
## TACC3              0          1.0792284         0.783         0.136
## DEPDC1             0          0.8344998         0.619         0.084
##        control_p_val_adj cardiomyopathy_p_val cardiomyopathy_avg_log2FC
## NCAPD2                 0                    0                 1.0627197
## ANLN                   0                    0                 2.1151653
## TACC3                  0                    0                 1.0922622
```

```
## DEPDC1                 0                   0                 0.8419712
##        cardiomyopathy_pct.1 cardiomyopathy_pct.2 cardiomyopathy_p_val_adj
## NCAPD2                0.809                0.177                        0
## ANLN                  0.983                0.197                        0
## TACC3                 0.764                0.111                        0
## DEPDC1                0.609                0.067                        0
##        max_pval minimump_p_val cluster
## NCAPD2        0              0       8
## ANLN          0              0       8
## TACC3         0              0       8
## DEPDC1        0              0       8
##
## [[10]]
##        control_p_val control_avg_log2FC control_pct.1 control_pct.2
## ANLN               0           1.526906         0.871         0.251
## DEPDC1             0           1.281879         0.632         0.093
## KIF20A             0           1.421288         0.628         0.051
## CENPF              0           2.411638         0.917         0.188
##        control_p_val_adj cardiomyopathy_p_val cardiomyopathy_avg_log2FC
## ANLN                   0                    0                  1.826956
## DEPDC1                 0                    0                  1.301962
## KIF20A                 0                    0                  1.399240
## CENPF                  0                    0                  2.725912
##        cardiomyopathy_pct.1 cardiomyopathy_pct.2 cardiomyopathy_p_val_adj
## ANLN                  0.872                0.199                        0
## DEPDC1                0.589                0.067                        0
## KIF20A                0.638                0.037                        0
## CENPF                 0.936                0.148                        0
##        max_pval minimump_p_val cluster
## ANLN          0              0       9
## DEPDC1        0              0       9
## KIF20A        0              0       9
## CENPF         0              0       9
##
## [[11]]
##        control_p_val control_avg_log2FC control_pct.1 control_pct.2
## 7SK    3.934583e-253          -2.478759         0.353         0.956
## ACTC1  1.564826e-200          -2.118641         0.388         0.908
## HSPB7  1.694370e-176          -1.821314         0.226         0.820
## MT-ND5 3.752436e-169          -1.408976         0.517         0.981
##        control_p_val_adj cardiomyopathy_p_val cardiomyopathy_avg_log2FC
## 7SK        1.596064e-248                    0                 -2.148121
## ACTC1      6.347717e-196                    0                 -2.194357
## HSPB7      6.873214e-172                    0                 -1.576412
## MT-ND5     1.522176e-164                    0                 -1.099682
##        cardiomyopathy_pct.1 cardiomyopathy_pct.2 cardiomyopathy_p_val_adj
## 7SK                   0.407                0.954                        0
## ACTC1                 0.450                0.814                        0
## HSPB7                 0.262                0.734                        0
## MT-ND5                0.512                0.961                        0
```

```
##             max_pval minimump_p_val cluster
## 7SK    3.934583e-253              0      10
## ACTC1  1.564826e-200              0      10
## HSPB7  1.694370e-176              0      10
## MT-ND5 3.752436e-169              0      10
##
## [[12]]
##       control_p_val control_avg_log2FC control_pct.1 control_pct.2
## FGR               0          1.0533705         0.357         0.001
## ABCB5             0          1.4148434         0.429         0.003
## ITGAL             0          0.8620775         0.321         0.001
## MATK              0          0.4812286         0.214         0.000
##       control_p_val_adj cardiomyopathy_p_val cardiomyopathy_avg_log2FC
## FGR                   0                    0                 1.0975151
## ABCB5                 0                    0                 0.7952365
## ITGAL                 0                    0                 1.1037954
## MATK                  0                    0                 0.4302709
##       cardiomyopathy_pct.1 cardiomyopathy_pct.2 cardiomyopathy_p_val_adj
## FGR                  0.476                0.001                        0
## ABCB5                0.230                0.004                        0
## ITGAL                0.462                0.002                        0
## MATK                 0.196                0.001                        0
##       max_pval minimump_p_val cluster
## FGR          0              0      11
## ABCB5        0              0      11
## ITGAL        0              0      11
## MATK         0              0      11
##
## [[13]]
##        control_p_val control_avg_log2FC control_pct.1 control_pct.2
## NCAPD2             0           1.557345         0.933         0.192
## ANLN               0           2.492987         0.992         0.258
## TACC3              0           1.753788         0.968         0.153
## DEPDC1             0           1.872563         0.925         0.094
##        control_p_val_adj cardiomyopathy_p_val cardiomyopathy_avg_log2FC
## NCAPD2                 0                    0                  1.523211
## ANLN                   0                    0                  2.520786
## TACC3                  0                    0                  1.751891
## DEPDC1                 0                    0                  1.810751
##        cardiomyopathy_pct.1 cardiomyopathy_pct.2 cardiomyopathy_p_val_adj
## NCAPD2                0.915                0.185                        0
## ANLN                  0.993                0.208                        0
## TACC3                 0.941                0.119                        0
## DEPDC1                0.886                0.071                        0
##        max_pval minimump_p_val cluster
## NCAPD2        0              0      12
## ANLN          0              0      12
## TACC3         0              0      12
## DEPDC1        0              0      12
##
```

```
## [[14]]
##         control_p_val control_avg_log2FC control_pct.1 control_pct.2
## EBF3     0.000000e+00           1.210283         0.299         0.010
## POU4F1   0.000000e+00           0.843252         0.309         0.014
## SHOX2    0.000000e+00           1.519801         0.419         0.009
## SLC24A3 2.533834e-215           1.962505         0.715         0.121
##         control_p_val_adj cardiomyopathy_p_val cardiomyopathy_avg_log2FC
## EBF3          0.00000e+00                    0                 0.6465859
## POU4F1        0.00000e+00                    0                 0.4779763
## SHOX2         0.00000e+00                    0                 0.6390848
## SLC24A3      1.02785e-210                    0                 1.3264197
##         cardiomyopathy_pct.1 cardiomyopathy_pct.2 cardiomyopathy_p_val_adj
## EBF3                   0.173                0.015                        0
## POU4F1                 0.156                0.004                        0
## SHOX2                  0.163                0.007                        0
## SLC24A3                0.479                0.114                        0
##              max_pval minimump_p_val cluster
## EBF3     0.000000e+00              0      13
## POU4F1   0.000000e+00              0      13
## SHOX2    0.000000e+00              0      13
## SLC24A3 2.533834e-215              0      13
##
## [[15]]
##        control_p_val control_avg_log2FC control_pct.1 control_pct.2
## H3C12  4.105655e-233          1.3199572         0.489         0.033
## H2AC17 1.466442e-202          1.0149642         0.382         0.023
## H1-5   3.687602e-150          2.2108662         0.511         0.058
## H2BC9  7.788989e-149          0.9395041         0.343         0.025
##        control_p_val_adj cardiomyopathy_p_val cardiomyopathy_avg_log2FC
## H3C12      1.665459e-228                    0                 1.3415217
## H2AC17     5.948621e-198                    0                 1.0970541
## H1-5       1.495876e-145                    0                 2.2977198
## H2BC9      3.159604e-144                    0                 0.9284186
##        cardiomyopathy_pct.1 cardiomyopathy_pct.2 cardiomyopathy_p_val_adj
## H3C12                 0.474                0.028                        0
## H2AC17                0.378                0.016                        0
## H1-5                  0.524                0.044                        0
## H2BC9                 0.341                0.019                        0
##             max_pval minimump_p_val cluster
## H3C12  4.105655e-233              0      14
## H2AC17 1.466442e-202              0      14
## H1-5   3.687602e-150              0      14
## H2BC9  7.788989e-149              0      14
##
## [[16]]
##         control_p_val control_avg_log2FC control_pct.1 control_pct.2
## SLC7A11  0.000000e+00           1.888533         0.693         0.015
## ALDH1L2 1.092406e-268           1.279723         0.545         0.018
## NIBAN1   1.595720e-89           1.565594         0.534         0.053
## WARS1    5.590288e-78           1.583875         0.705         0.112
```

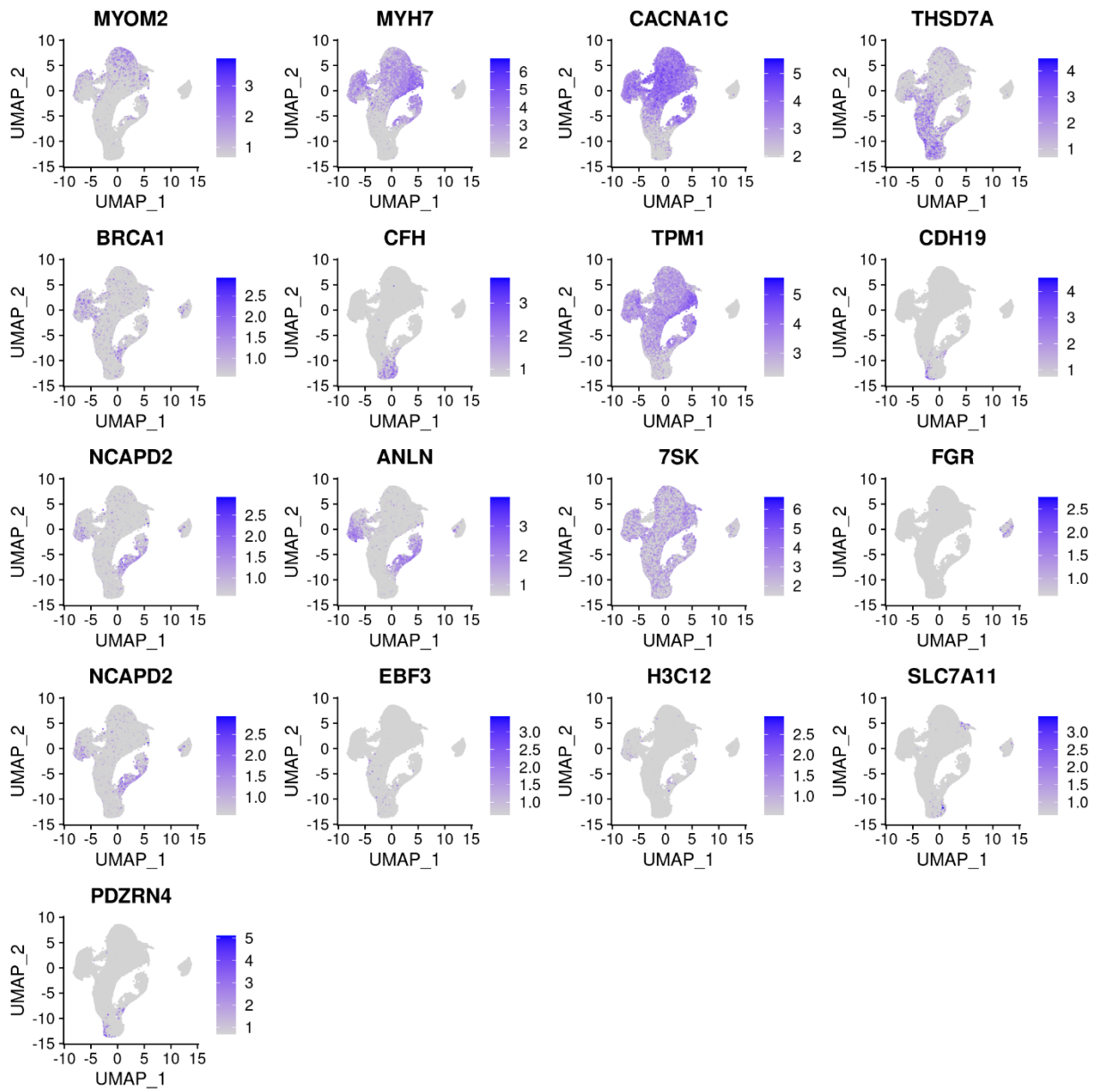
```
##         control_p_val_adj cardiomyopathy_p_val cardiomyopathy_avg_log2FC
## SLC7A11      0.000000e+00                    0                   1.744408
## ALDH1L2     4.431345e-264                    0                   1.331168
## NIBAN1       6.473037e-85                    0                   1.433293
## WARS1        2.267700e-73                    0                   1.359551
##         cardiomyopathy_pct.1 cardiomyopathy_pct.2 cardiomyopathy_p_val_adj
## SLC7A11                 0.61                0.022                        0
## ALDH1L2                 0.58                0.023                        0
## NIBAN1                  0.59                0.082                        0
## WARS1                   0.73                0.135                        0
##              max_pval minimump_p_val cluster
## SLC7A11  0.000000e+00              0      15
## ALDH1L2 1.092406e-268              0      15
## NIBAN1   1.595720e-89              0      15
## WARS1    5.590288e-78              0      15
##
## [[17]]
##        control_p_val control_avg_log2FC control_pct.1 control_pct.2
## PDZRN4 5.851587e-207           3.178166         0.590         0.027
## MXRA5  2.879015e-197           1.631970         0.602         0.029
## CDH19  2.370155e-194           2.228781         0.602         0.029
## MGP    5.103816e-188           1.151184         0.566         0.026
##        control_p_val_adj cardiomyopathy_p_val cardiomyopathy_avg_log2FC
## PDZRN4     2.373696e-202                    0                   2.667386
## MXRA5      1.167873e-192                    0                   1.355275
## CDH19      9.614533e-190                    0                   1.564279
## MGP        2.070363e-183                    0                   1.172662
##        cardiomyopathy_pct.1 cardiomyopathy_pct.2 cardiomyopathy_p_val_adj
## PDZRN4                0.371                0.025                        0
## MXRA5                 0.483                0.052                        0
## CDH19                 0.340                0.024                        0
## MGP                   0.508                0.057                        0
##              max_pval minimump_p_val cluster
## PDZRN4 5.851587e-207              0      16
## MXRA5  2.879015e-197              0      16
## CDH19  2.370155e-194              0      16
## MGP    5.103816e-188              0      16
```

```
# > top.markers
top.markers <- c("MYOM2", "MYH7","CACNA1C","THSD7A", "BRCA1", "CFH",
"TPM1","CDH19", "NCAPD2", "ANLN","7SK", "FGR", "NCAPD2", "EBF3","H3C12",
"SLC7A11", "PDZRN4")


# Cluster Gene Cell_type
#0 MYOM2="Cardiomyocyte"
#1 MYH7="Cardiomyocyte"
#2 CACNA1C="Cardiomyocyte"
#3 THSD7A="Neuroendocrine" ## **
#4 BRCA1="?Epithelial_cell"
```

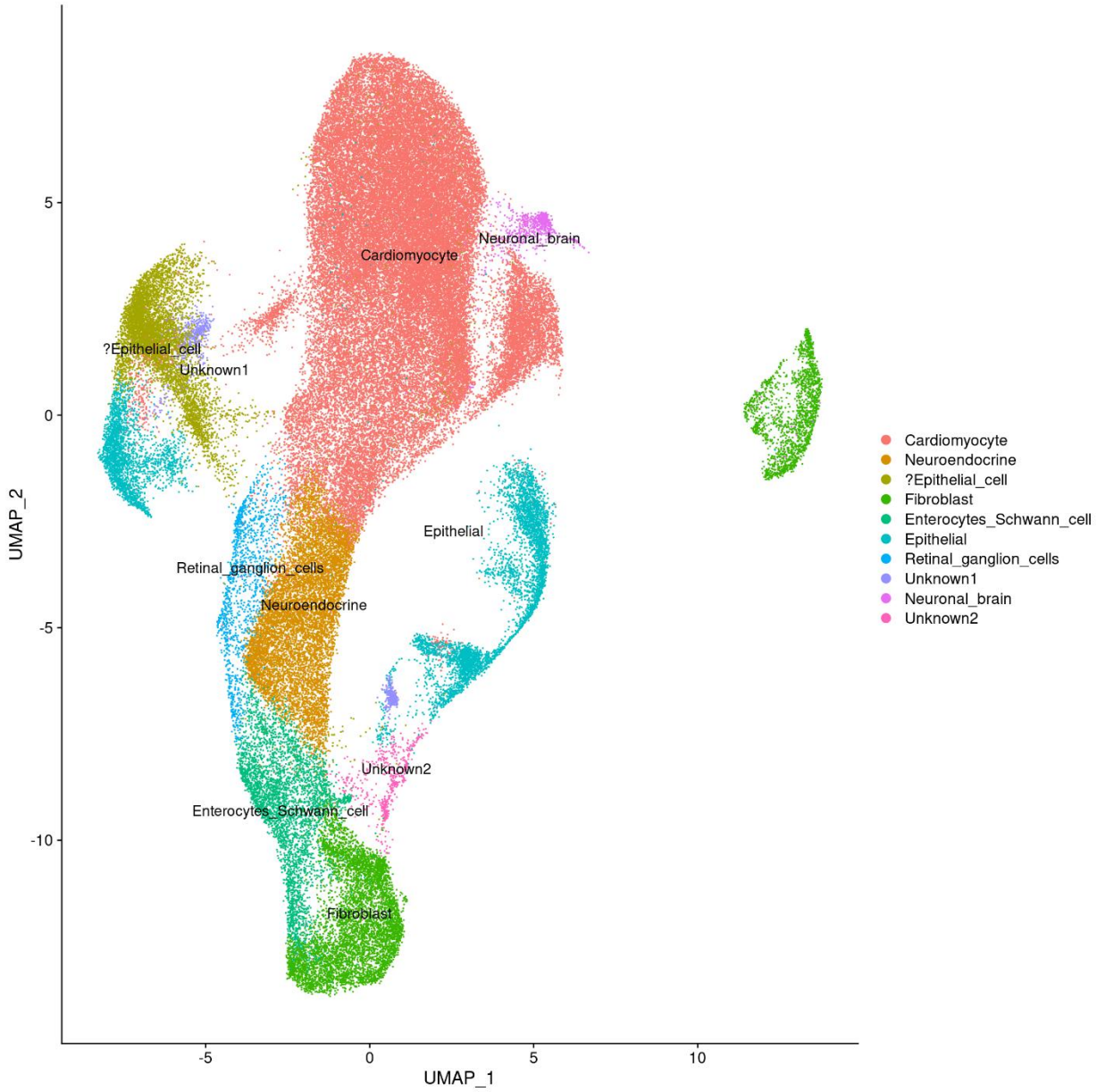
```
#5 CFH="Fibroblast"
#6 TPM1="Cardiomyocyte"
#7 CDH19="Enterocytes_Schwann_cell"
#8 ANLN="Epithelial"
#9 ANLN="Epithelial"
#10 ACTC1="Cardiomyocyte"
#11 FGR="Fibroblast"
#12 ANLN="Epithelial"
#13 POU4F1="Retinal_ganglion_cells"
#14 H3C12="Unknown1"
#15 SLC7A11="Neuronal_brain"
#16 PDZRN4="Unknown2"
```

Here we can use these marker genes from each cluster and use them to annotate our clusters as specific cell types.

```
DefaultAssay(immune.combined) <- "RNA"
p1 <- FeaturePlot(immune.combined, features = top.markers, min.cutoff = "q9")

SaveFigure(p1,"dimplot_individual_markers_CMP_vs_control", width = 12, height
= 12, res = 300)
```



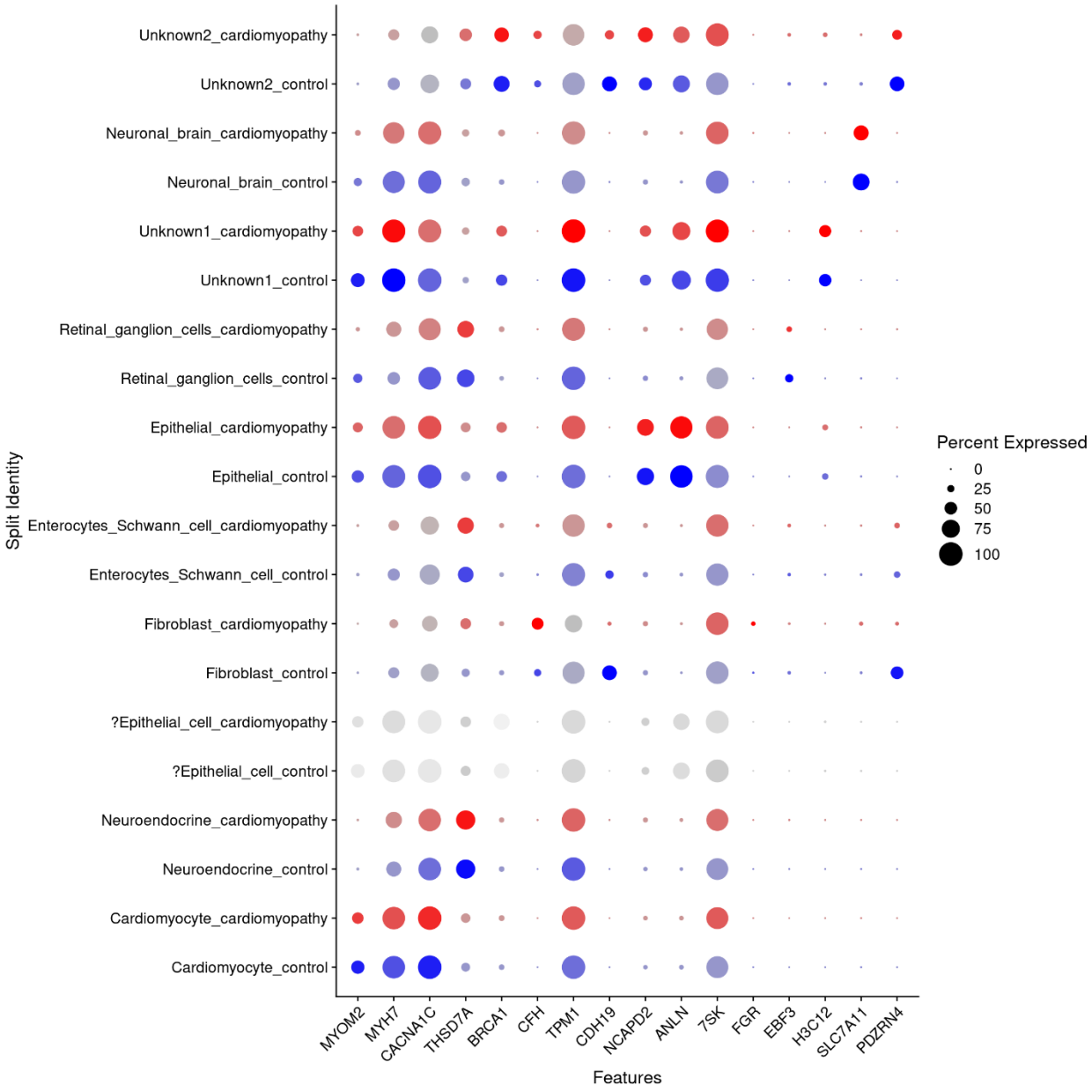

```
immune.combined <- RenameIdents(immune.combined, `0` = "Cardiomyocyte", `1` =
"Cardiomyocyte", `2` = "Cardiomyocyte", `3` = "Neuroendocrine", `4` =
"?Epithelial_cell", `5` = "Fibroblast", `6` = "Cardiomyocyte", `7` =
"Enterocytes_Schwann_cell", `8` = "Epithelial",
    `9` = "Epithelial", `10` = "Cardiomyocyte", `11` = "Fibroblast", `12` =
"Epithelial", `13` = "Retinal_ganglion_cells", `14` = "Unknown1",
    `15` = "Neuronal_brain", `16` = "Unknown2")
p1 <- DimPlot(immune.combined, label = TRUE, repel = TRUE)
SaveFigure(p1,"dimplot_annotated_CMP_vs_control", width = 12, height = 12,
res = 300)
```



The DotPlot() function with the split.by parameter can be useful for viewing conserved cell type markers across conditions, showing both the expression level and the percentage of cells in a cluster expressing any given gene. Here we plot 2-3 strong marker genes for each of our 14 clusters.

```
markers.to.plot <- unique(c(top.markers))
p1 <- DotPlot(immune.combined, features = markers.to.plot, cols = c("blue",
"red"), dot.scale = 8, split.by = "grouping_var1") +
    RotatedAxis()
```

```
SaveFigure(p1,"dotplot_top_markers_CMP_vs_control", width = 12, height = 12,
res = 300)
```

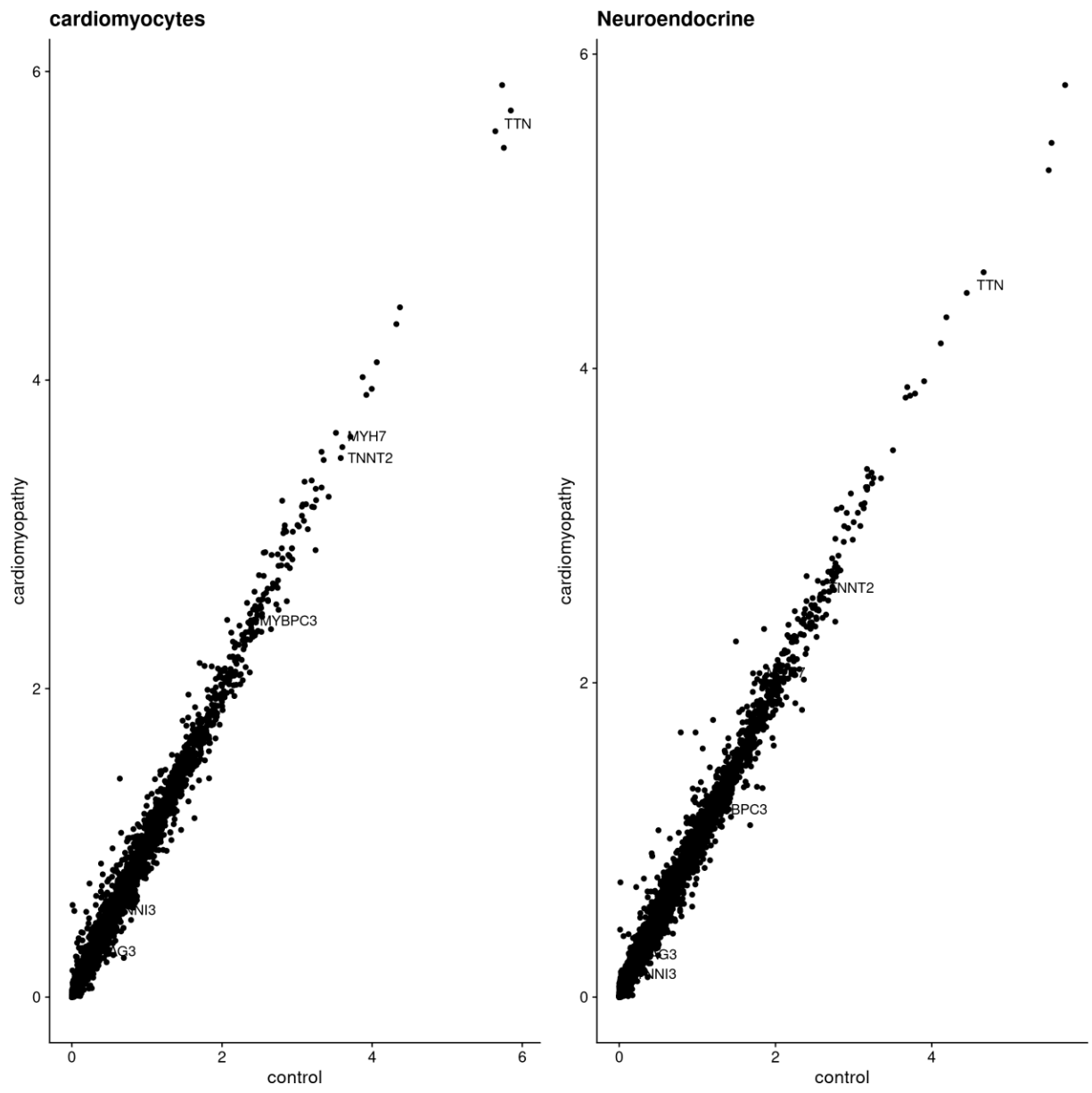


Identify differential expressed genes across conditions

Now that we've aligned the stimulated and control cells, we can start doing comparative analyses and look at the differences induced by stimulation. One way to look broadly at these changes is to plot the average expression of both the stimulated and control cells and look for genes that are visual outliers on a scatter plot. Here, we take the average

expression of both the stimulated and control Cardiomyocytes and Neuroendocrine populations and generate the scatter plots, highlighting genes that exhibit dramatic responses.

```
library(ggplot2)
library(cowplot)
theme_set(theme_cowplot())
Cardiomyocytes <- subset(immune.combined, idents = "Cardiomyocyte")
Idents(Cardiomyocytes) <- "grouping_var1"
avg.Cardiomyocytes <- as.data.frame(log1p(AverageExpression(Cardiomyocytes,
verbose = FALSE)$RNA))
avg.Cardiomyocytes$gene <- rownames(avg.Cardiomyocytes)

Neuroendocrine <- subset(immune.combined, idents = "Neuroendocrine")
Idents(Neuroendocrine) <- "grouping_var1"
avg.Neuroendocrine <- as.data.frame(log1p(AverageExpression(Neuroendocrine,
verbose = FALSE)$RNA))
avg.Neuroendocrine$gene <- rownames(avg.Neuroendocrine)

# genes.to.label = c("ISG15", "LY6E", "IFI6", "ISG20", "MX1", "IFIT2",
"IFIT1", "CXCL10", "CCL8")
genes.to.label = c("TTN", "BAG3", "MYH7", "MYBPC3", "TNNT2", "TNNI3", "LMNA",
"DSP", "RYR2", "ACTC1", "PLN")
p1 <- ggplot(avg.Cardiomyocytes, aes(control, cardiomyopathy)) + geom_point()
+ ggtitle("cardiomyocytes")
p1 <- LabelPoints(plot = p1, points = genes.to.label, repel = TRUE)
p2 <- ggplot(avg.Neuroendocrine, aes(control, cardiomyopathy)) + geom_point()
+ ggtitle("Neuroendocrine")
p2 <- LabelPoints(plot = p2, points = genes.to.label, repel = TRUE)
p1 + p2
SaveFigure((p1 +
p2),"differentially_expressed_genes_across_clusters_CMP_vs_control", width =
12, height = 12, res = 300)
```



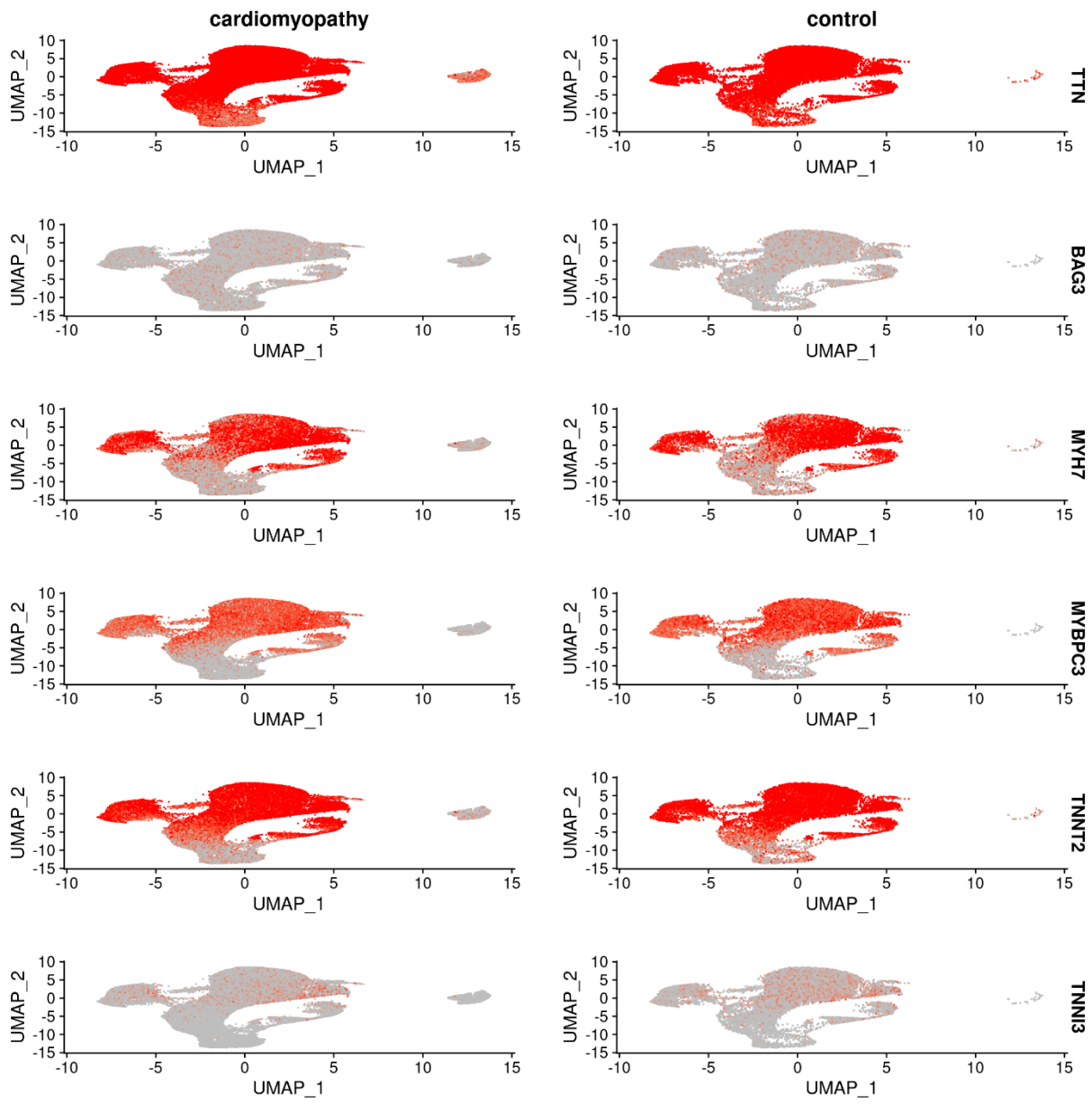
As we can see, many of the same genes (genes selected for cardiomyopathy) are upregulated in both of these cell types and likely represent a conserved pathway.

Because we are confident in having identified common cell types across condition, we can ask what genes change in different conditions for cells of the same type. First, we create a column in the meta.data slot to hold both the cell type and stimulation information and switch the current ident to that column. Then we use FindMarkers() to find the genes that are different between cardiomyopathy and control cardiomyocyte cells. This can be explored for other cell clusters as well.

```
immune.combined$celltype.cardiomyopathy <- paste(Idents(immune.combined),
immune.combined$grouping_var1, sep = "_")
immune.combined$celltype <- Idents(immune.combined)
Idents(immune.combined) <- "celltype.cardiomyopathy"
trt_control.markers <- FindMarkers(immune.combined, ident.1 =
"Cardiomyocyte_cardiomyopathy", ident.2 = "Cardiomyocyte_control", verbose =
FALSE)
head(trt_control.markers, n = 15)

# SaveObject(trt_control.markers, "cmp_vs_control.DGE")
# ReadObject("cmp_vs_control.DGE")
write.table(trt_control.markers,
"cardiomyopathy_Vs_control_DGE_in_cardiomyocyte_cluster.txt", col.names = T,
row.names = T, quote = F)
```

Another useful way to visualize these changes in gene expression is with the split.by option to the FeaturePlot() or VlnPlot() function. This will display FeaturePlots of the list of given genes, split by a grouping variable (stimulation condition here).

```
p1 <- FeaturePlot(immune.combined, features = genes.to.label, split.by =
"grouping_var1", max.cutoff = 3,
    cols = c("grey", "red"))

SaveFigure(p1,"DGE_feature_plot_CMP_vs_control", width = 12, height = 12, res
= 300)
```



Sources used for cluster identification:
GTex: https://gtexportal.org/home/multiGeneSingleCellQueryPage
PanglaoDB: https://panglaodb.se/markers.html?cell_type=%27all_cells%27