scRNA-seq Analysis

Benedict

03/03/2021

Single Cell RNA-seq Analysis with Seurat

This R Markdown document performs differential gene expression testing for a group of samples.

Project Structure

```
project
    README.md
  data

      RNA
          N2
          B2
          E2
          M2
      ATAC
          X31-OVA_DIV0
          X31-OVA_DIV3
  code
     differential_scRNA-seq.Rmd
     differential_scATAC-seq.Rmd
  results
     RNA
       scRNA-seq-DE.xlsx
       expression_heatmap.png
       expression_dotplot.png
     ATAC
          A
          B
```

Dependencies

- tidyverse
- Seurat
- cowplot
- biomart
- openxlsx
- patchwork

Data Import

Uncomment following code block to analyse the DIV0-4 samples

```
# data_folder <- "../data/united/RNA/"

# div_0.data <-  Read10X(data.dir = paste0(data_folder,
#                                          "X31-OVA-D0/filtered_feature_bc_matrix"))
# div_1.data <-  Read10X(data.dir = paste0(data_folder,
#                                          "X31-OVA-D1/filtered_feature_bc_matrix"))
# div_2.data <-  Read10X(data.dir = paste0(data_folder,
#                                          "X31-OVA-D2/filtered_feature_bc_matrix"))
# div_3.data <-  Read10X(data.dir = paste0(data_folder,
#                                          "X31-OVA-D3/filtered_feature_bc_matrix"))
# div_4.data <-  Read10X(data.dir = paste0(data_folder,
#                                          "X31-OVA-D4/filtered_feature_bc_matrix"))
# samples <- c(div_0.data, div_1.data, div_2.data, div_3.data, div_4.data)
# names <- c("DIV0", "DIV1", "DIV2", "DIV3", "DIV4)
```

Uncomment following code block to analyse the N2, B2, E2, M2 samples

```
data_folder <- "../data/united/RNA/"
N2.data <- Read10X(data.dir = paste0(data_folder,
                                     "N2/filtered_feature_bc_matrix"))
B2.data <- Read10X(data.dir = paste0(data_folder,
                                     "B2/filtered_feature_bc_matrix"))
E2.data <- Read10X(data.dir = paste0(data_folder,
                                     "E2/filtered_feature_bc_matrix"))
M2.data <- Read10X(data.dir = paste0(data_folder,
                                     "M2/filtered_feature_bc_matrix"))
samples <- c(N2.data, B2.data, E2.data, M2.data)
names <- c("N2", "B2", "E2", "M2")
```

Create Seurat objects from input data

```
sample_objs <- c()
idx = 1
for(sample in samples){
 sample <- CreateSeuratObject(counts = sample,
                      project = "CaseStudy2021",
                      min.cells = 3,
                      min.features = 200)
 sample@meta.data$state <- names[idx]
 sample_objs <- append(sample_objs, sample)
 idx = idx + 1
}
```

Compute QC metrics

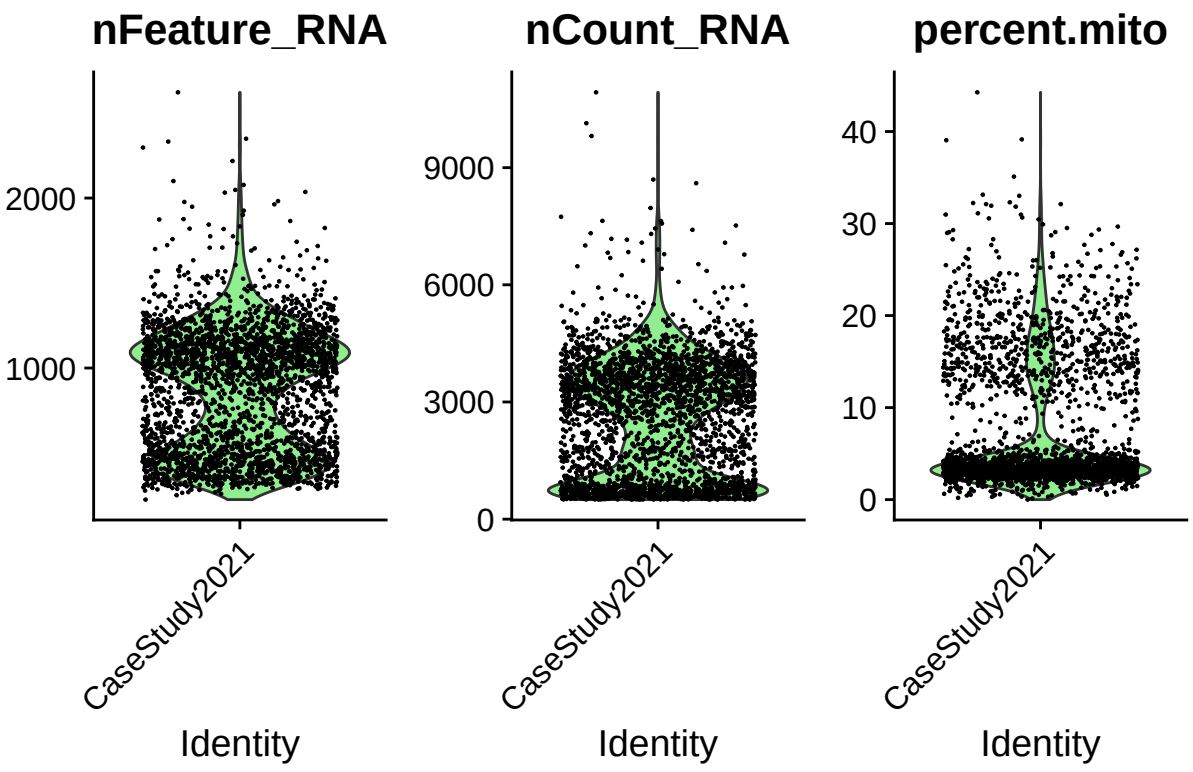
Workflow computes and applied the following commonly used metrics in a QC step:

- Number of unique genes detected in each cell
  - Low-quality cells or empty droplets often have very few genes
  - Cell multiplets may have an abnormally high gene count
- Total number of molecules detected within a cell (should correlate with first metric)
- Percentage of reads mapping to the mitochondrial genome
- Low-quality and dying cells often contain high mitochondrial contamination

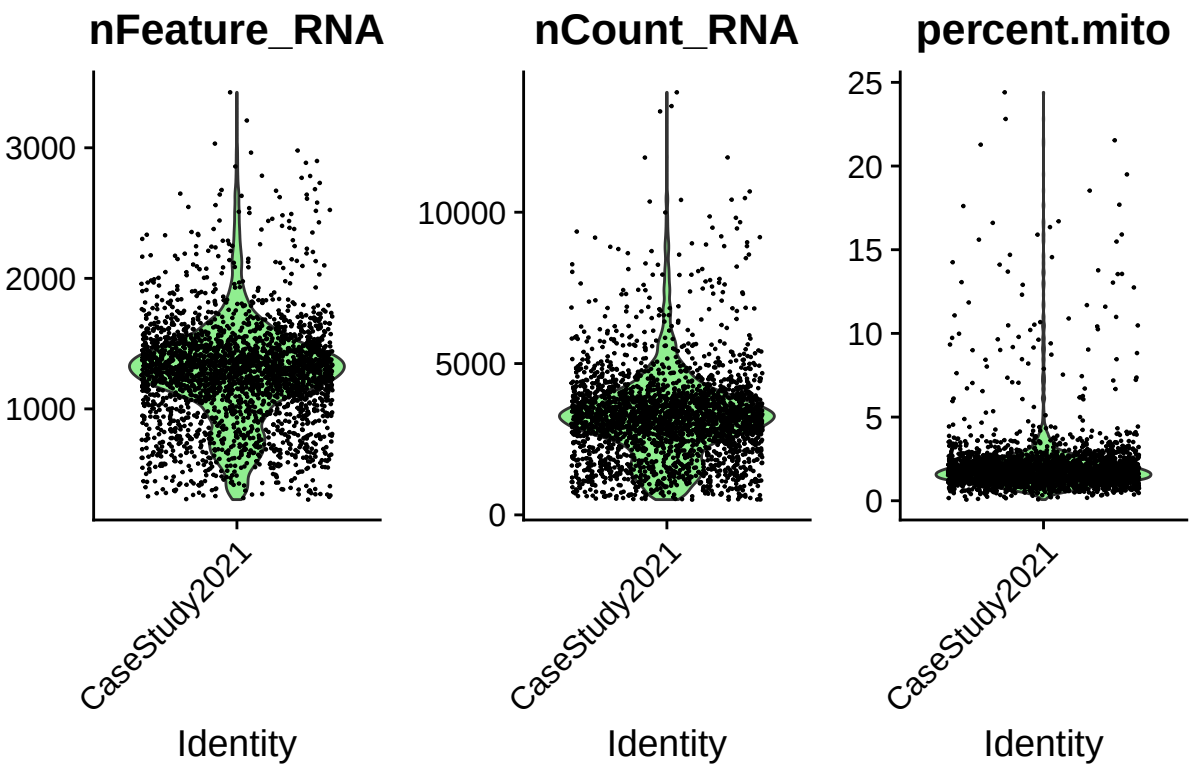
```
temp <- c()
for(seurat_obj in sample_objs){
  mito.genes <- rownames(seurat_obj)[grep("^mt-",rownames(seurat_obj))]
  c <- GetAssayData(object = seurat_obj, slot = "counts")
  percent.mito <- colSums(c[mito.genes,])/Matrix::colSums(c)*100
  seurat_obj$percent.mito <- percent.mito
  temp <- append(temp, seurat_obj)
}
sample_objs <- temp
rm(temp)
```

```
plots <- list()
for (sample_obj in sample_objs){
  plot(VlnPlot(object = sample_obj,
               features = c("nFeature_RNA", "nCount_RNA", "percent.mito"),
               cols = c('light green'),
               combine = TRUE) +
         plot_annotation(
           title = paste0(sample_obj@meta.data$state[[1]], " QC Metrics")
         ))
  }
```

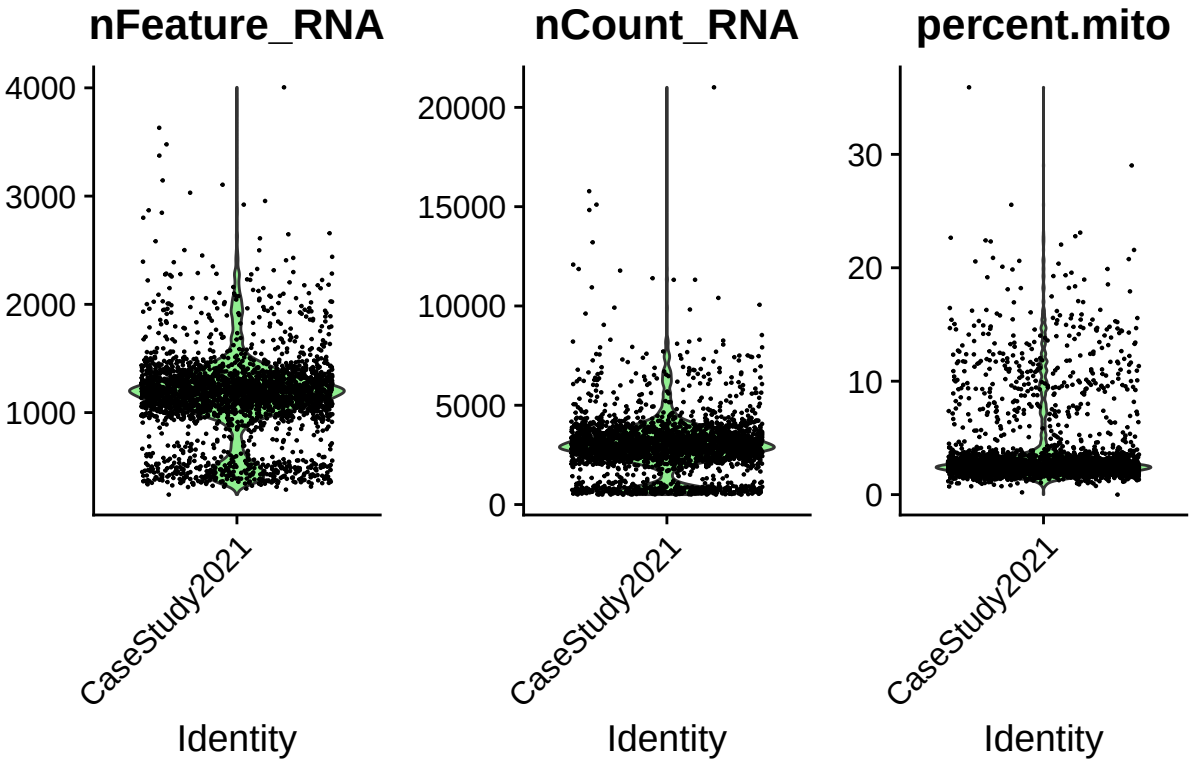
N2 QC Metrics

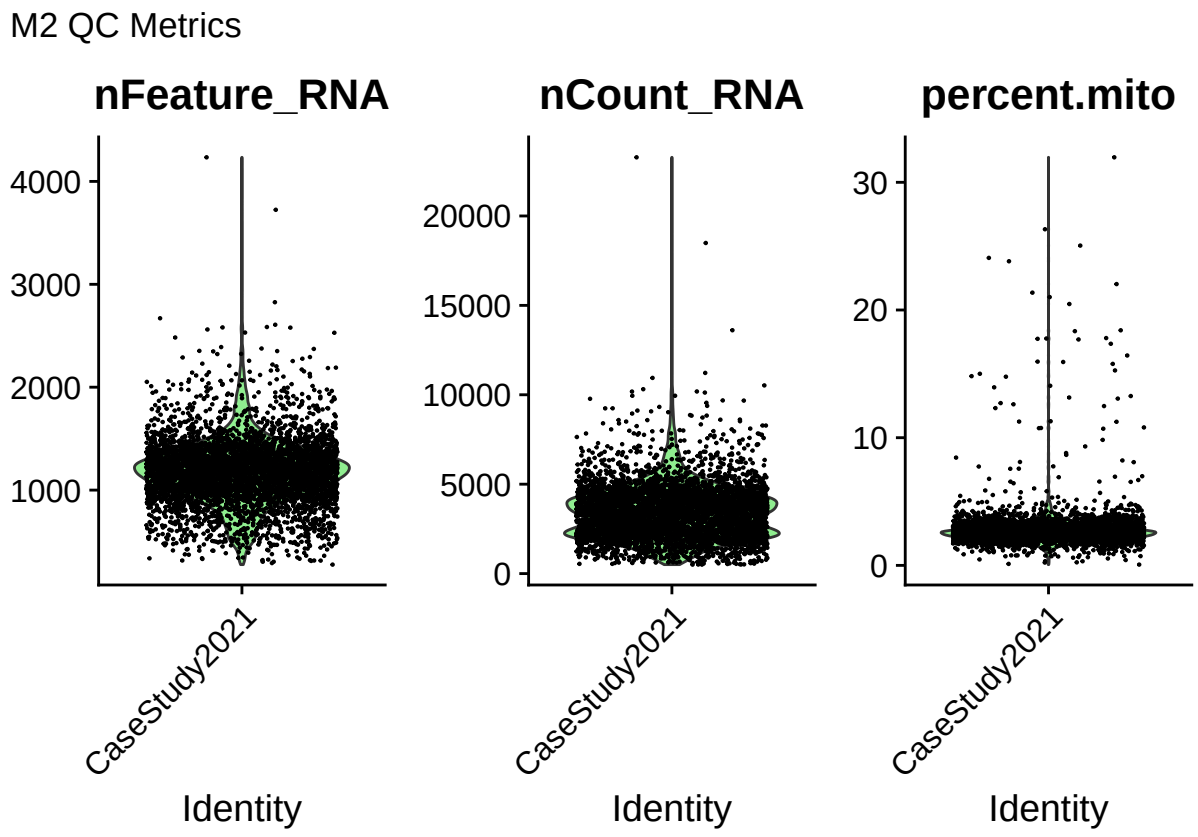


B2 QC Metrics



E2 QC Metrics





Filter cells based on QC metrics

Normalize Samples

The data is normalized using a global-scaling normalization method to normalise gene expression levels for each cell by the total expression. This is multipliplied the a scale factor of 1000 and the result is natural-log-transformed.

```
sample_objs <- lapply(X = sample_objs, FUN = function(x) {
    x <- NormalizeData(x, normalization.method="LogNormalize", scale.factor=1000)
    x <- FindVariableFeatures(x, selection.method = "vst", nfeatures = 2000)
})
```

Merge Samples

```
samples.combined <- merge(sample_objs[[1]], y = sample_objs[-1],
                          add.cell.ids =names, project = "CaseStudy2021")
```

Scale Data

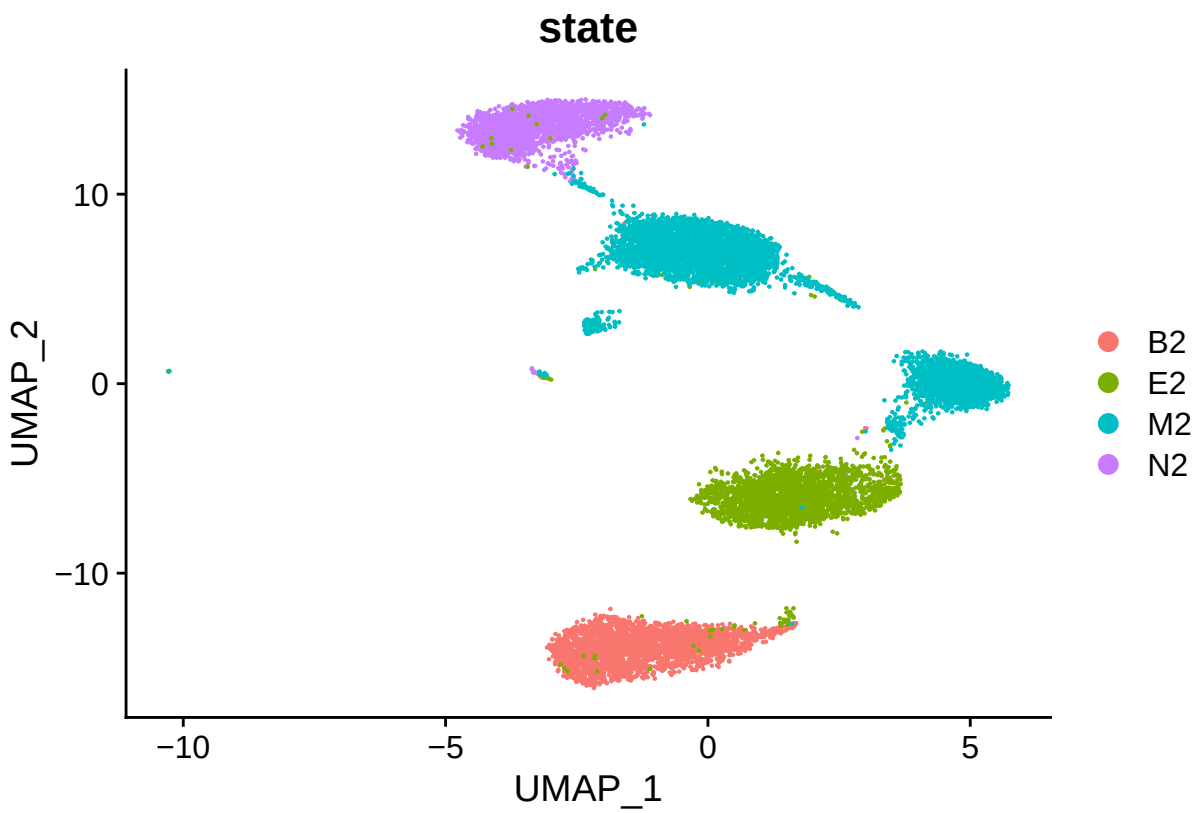
Pre-processing step before dimensional reduction is performed

- Expression of each gene shifted to give mean expression of 0 across cells
- Expression of each gene scaled to give a variance of 1 across cells
  - Prevents domination from highly-expressed genes

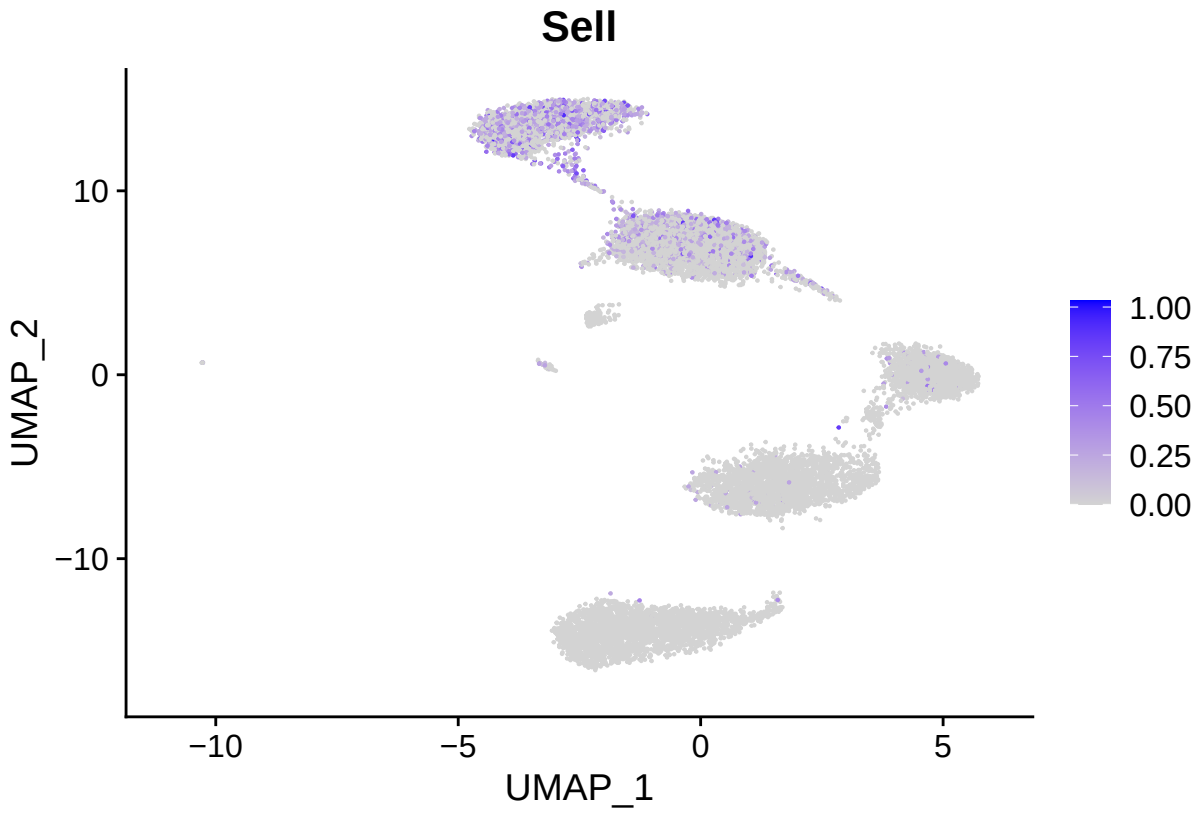
Linear Dimensionality Reduction

UMAP Plot

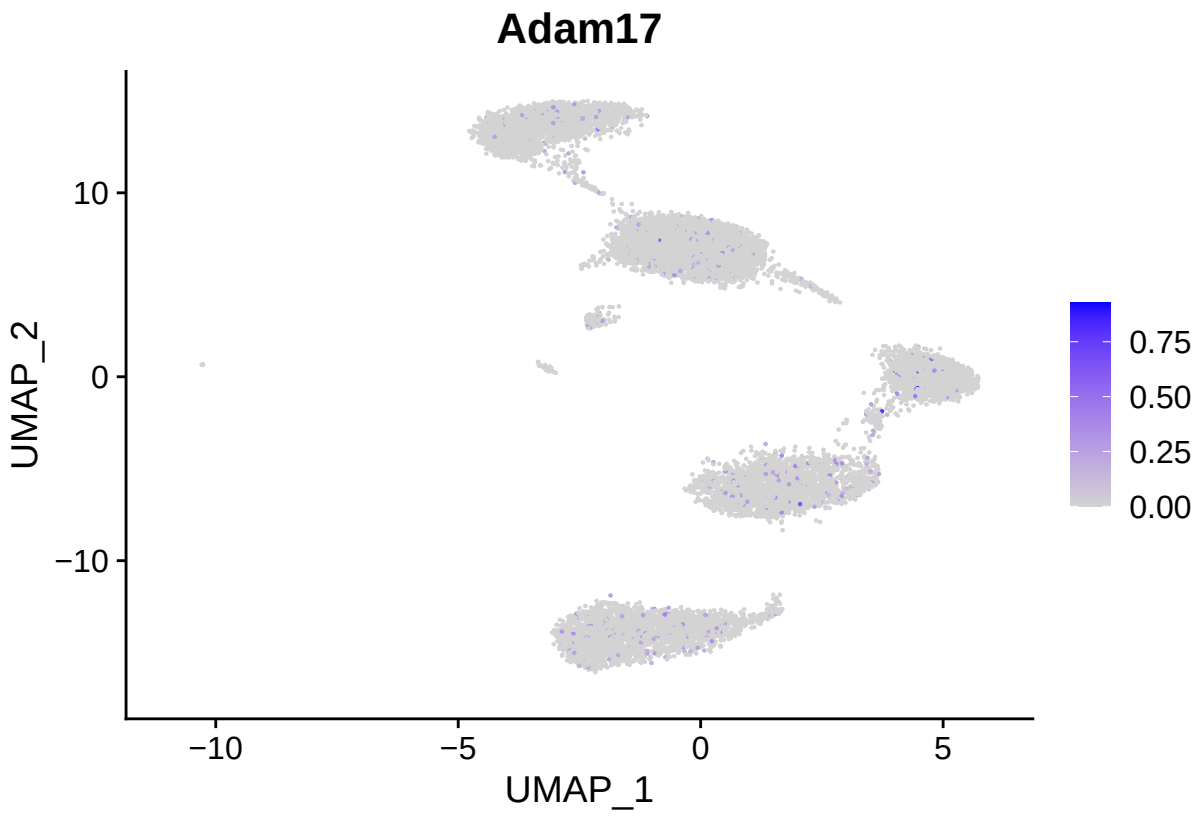
```
plt <- DimPlot(samples.combined, reduction = "umap", group.by = "state")
plt
```



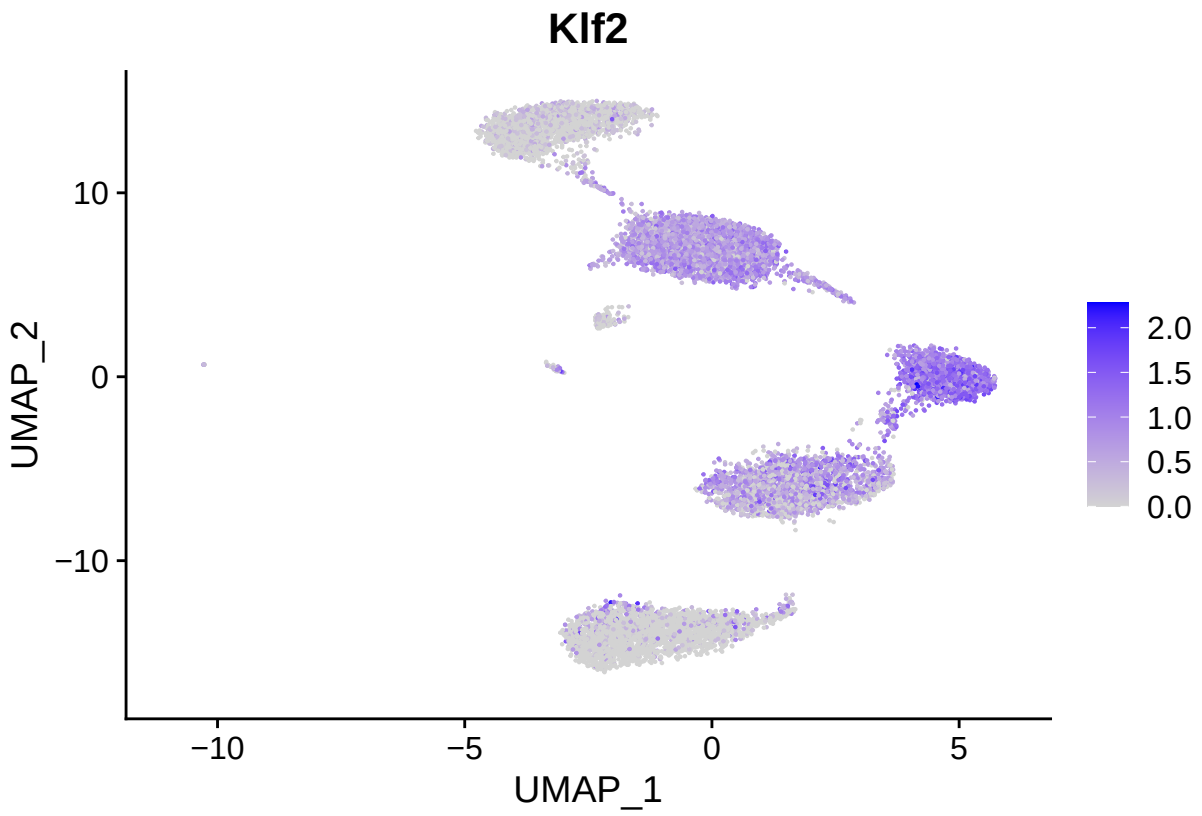
```
plt <- FeaturePlot(samples.combined, features = c("Sell"))
plt
```



```
plt <- FeaturePlot(samples.combined, features = c("Adam17"))
plt
```



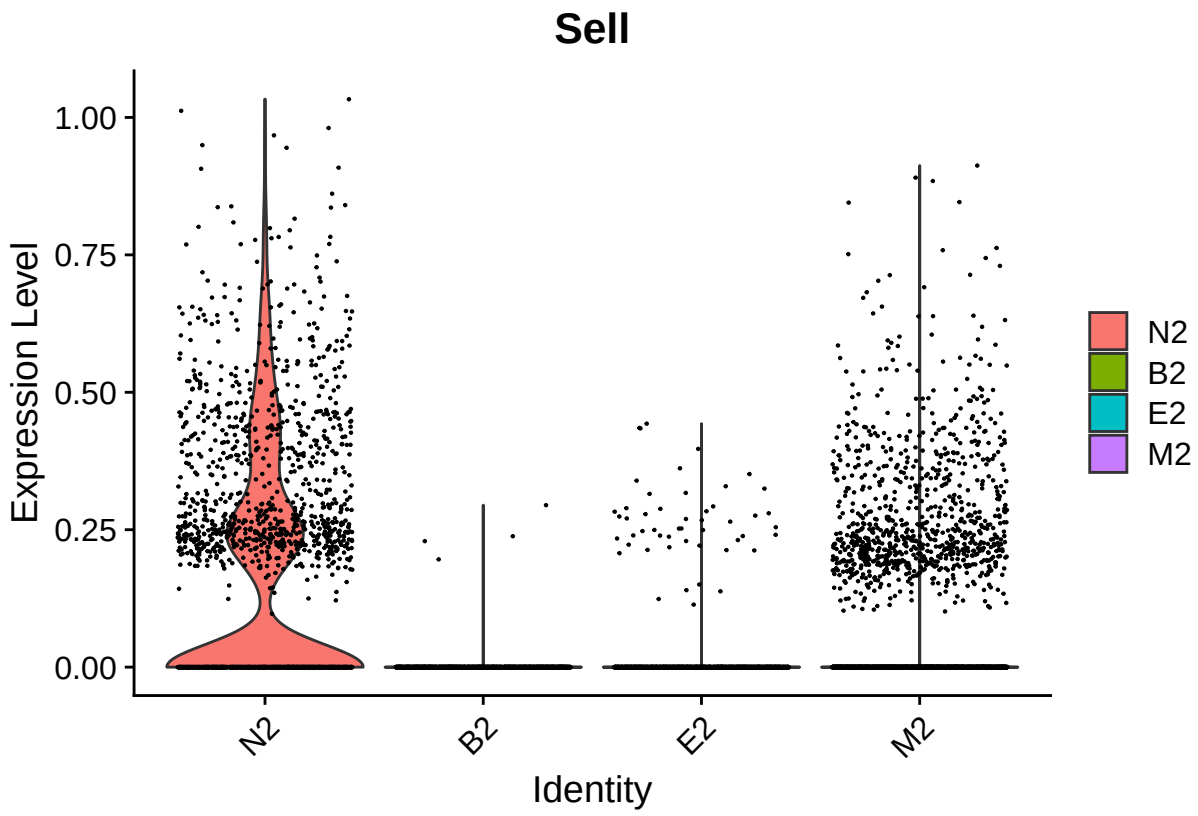
```
plt <- FeaturePlot(samples.combined, features = c("Klf2"))
plt
```



Expression Violin Plots

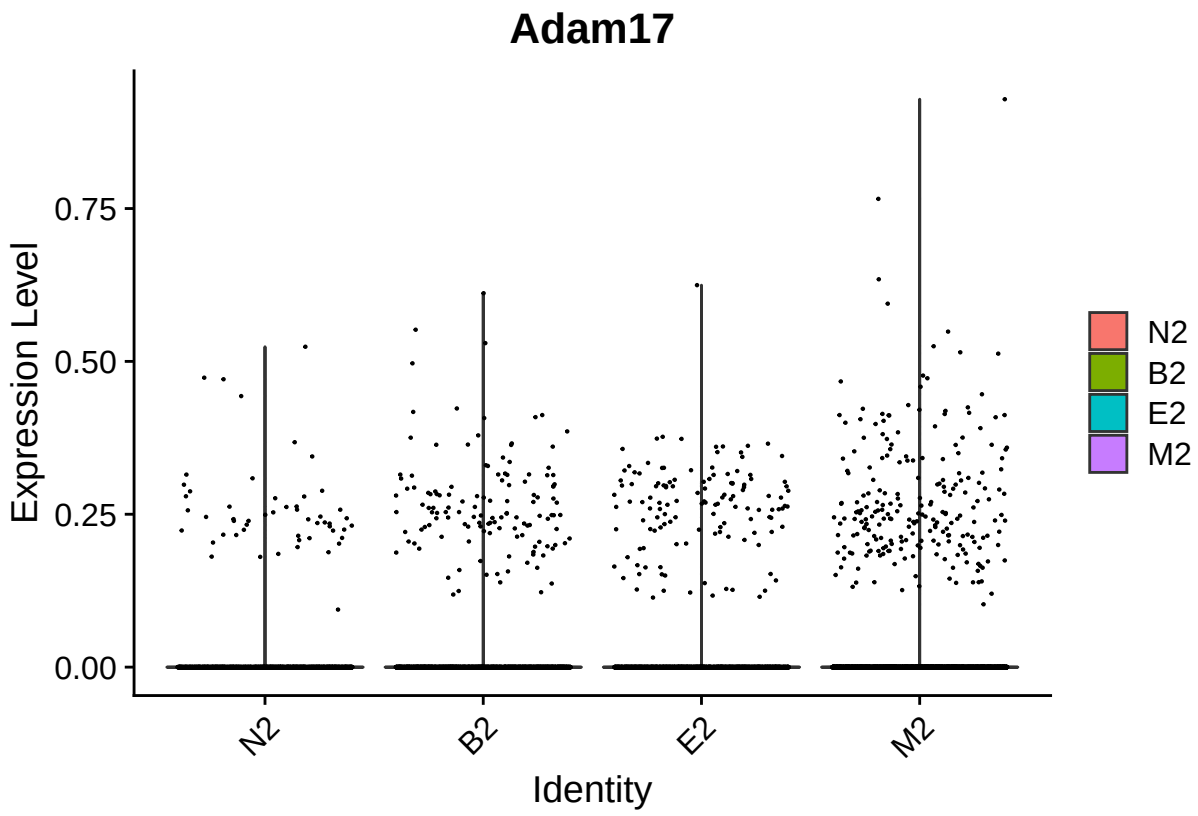
Sell

```
Idents(samples.combined) <- "state"
VlnPlot(samples.combined, features = c("Sell"))
```



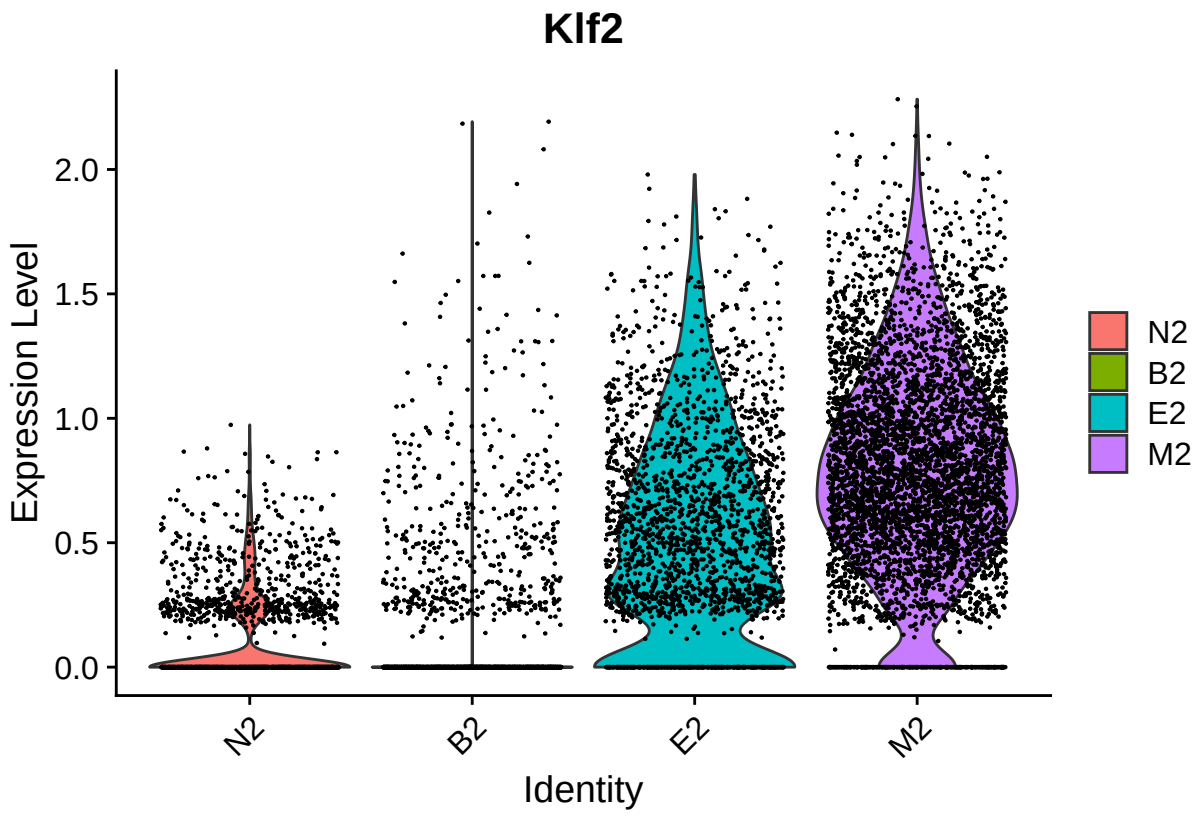
Adam17

```
VlnPlot(samples.combined, features = c("Adam17"))
```



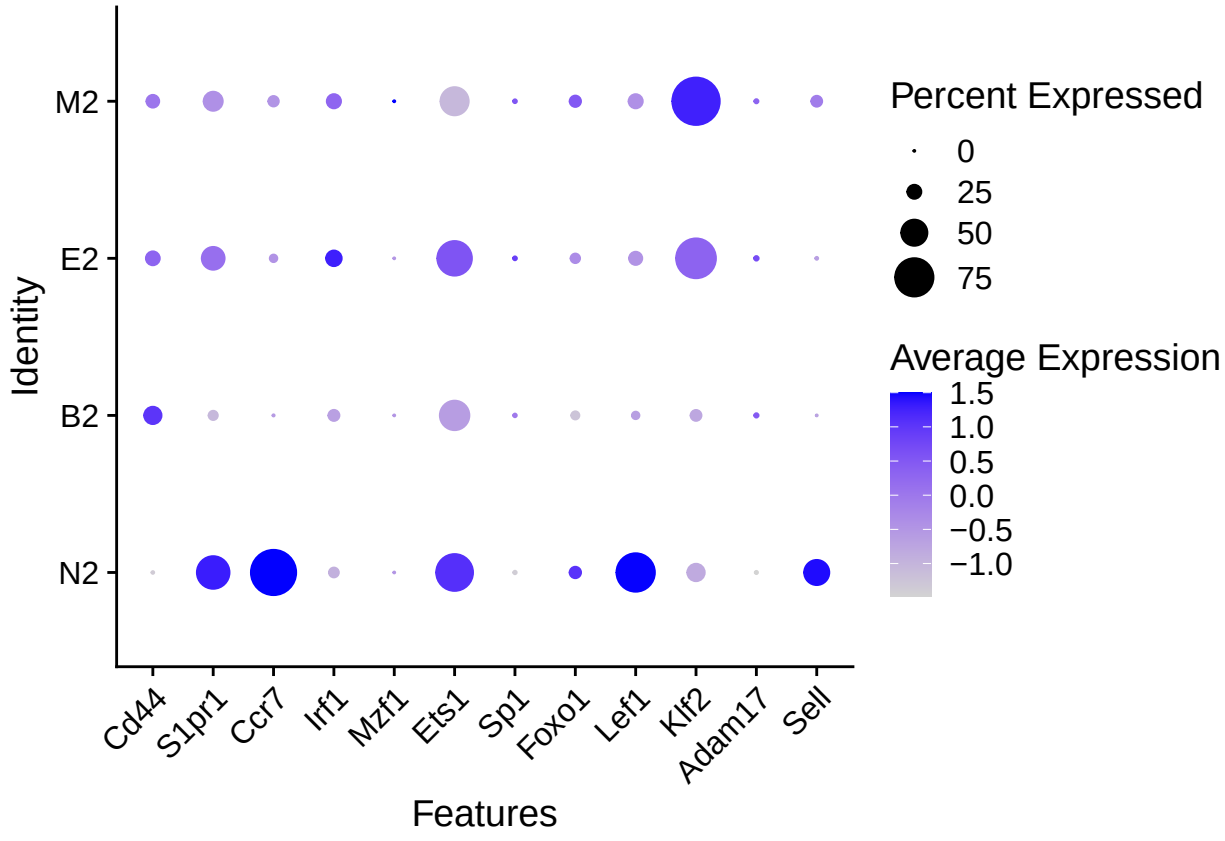
Klf2

```
VlnPlot(samples.combined, features = c("Klf2"))
```


Klf2
Expression Level
2.0
1.5
1.0
0.5
0.0
N2
B2
E2
M2
Identity
N2
B2
E2
M2

Expression Dot Plot

```
genes_of_note <- c("Sell", "Adam17", "Klf2", "Lef1", "Foxo1", "Sp1", "Ets1",
                   "Mzf1", "Irf1", "Ccr7", "S1pr1", "Cd44")
plt <- DotPlot(samples.combined, features = rev(genes_of_note), dot.scale = 8) +
  RotatedAxis()
save_plot("../results/RNA/expression_dotplot.png", plt)
plt
```



Expression Heatmap

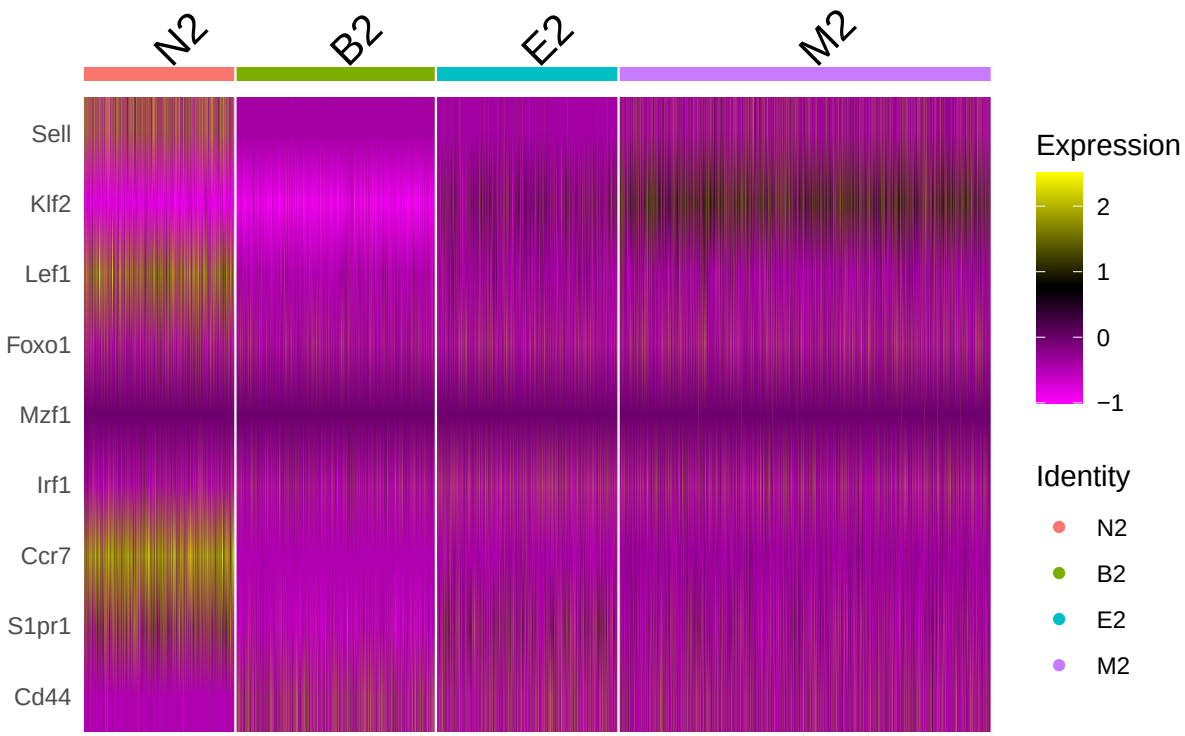
```
samples.combined <- ScaleData(samples.combined)
```

```
## Centering and scaling data matrix
```

```
plt <- DoHeatmap(samples.combined, features=genes_of_note)
```

```
## Warning in DoHeatmap(samples.combined, features = genes_of_note): The following
## features were omitted as they were not found in the scale.data slot for the RNA
## assay: Ets1, Sp1, Adam17
```

```
save_plot("../results/RNA/expression_heatmap.png", plt)
plt
```



Perform Differential Expression (DE) Testing

Compare expression betwen markers

Performs a Wilcoxon Rank Sum test to identify differentially expressed genes between a sample and all other cells, for each sample

Annotate DE genes from ensembl

```
ensembl <- useEnsembl(biomart = "ensembl", dataset = "mmusculus_gene_ensembl")
genes <- unique(samples.combined.markers$gene)
genedesc <- getBM(attributes=c('external_gene_name',
                               'description',
                               'chromosome_name',
                               'start_position',
                               'end_position'),
                  filters = 'external_gene_name',
                  values = genes,
                  mart =ensembl)

samples.combined.markers <- merge(samples.combined.markers, genedesc, all=TRUE,
                                  by.x = c("gene"), by.y = c("external_gene_name"))

# Reorder columns
new_order <- c("gene", "cluster", "avg_log2FC", "p_val_adj", "pct.1", "pct.2",
               "description", "chromosome_name", "start_position", "end_position")
samples.combined.markers <- samples.combined.markers[new_order]
```

Write to excel spreadsheet 'scRNA-seq-DE'

Results of differential gene expression test are written to an excel workbook with two sheets. Sheet one is all results with genes of particular importance being highlighted in yellow Sheet two contains only those genes of particular importance

```
excel <- createWorkbook("scRNA-seq-DE")
addWorksheet(excel, "scRNA-seq")
writeData(excel, sheet = 1, samples.combined.markers)

# Highlight genes of particular interest
samples.combined.markers$rownum <- 1:nrow(samples.combined.markers)
rw <- samples.combined.markers %>% filter(gene %in% genes_of_note )
highlight_style <- openxlsx::createStyle(fgFill = "yellow")
openxlsx::addStyle(wb = excel,
                   sheet = 1,
                   style = highlight_style,
                   rows = (rw$rownum+1),
                   cols = 1:ncol(rw),
                   stack = TRUE,
                   gridExpand = TRUE)

# Create second sheet containing only genes of note
```

```r
rw$rownum <- NULL
rw <- rw %>% arrange(gene)
addWorksheet(excel, "scRNA-seq-specific")
writeData(excel, sheet=2, rw)

# Finally write out to file
saveWorkbook(excel, file = "../results/RNA/scRNA-seq-DE.xlsx", overwrite = TRUE)
```