Introduction to Single-cell RNA-seq Analysis - 4

by Jiajia Li

Research School of Biology

27 Aug 2025

Australian
National
University

Learning Objectives of Today

- Single-cell Differential Expression Analysis on epithelial cells
- Pseudobulk Differential Expression Analysis on epithelial cells

- Both analyses on T cells.
- Also on CD8 T cells.

Differential expression analysis

In this lesson we will use the previously generated Seurat object for **gene expression** and **differential expression analyses**.

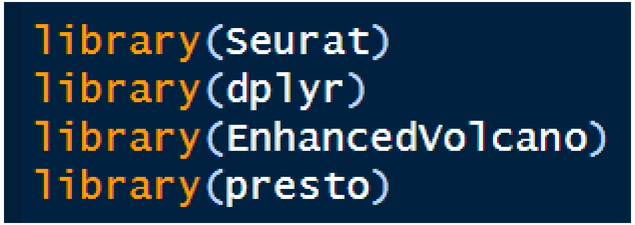
We will carry out two sets of differential expression analyses.

Firstly, since we know that the tumour cells should be epithelial cells, we will begin by trying to identify epithelial cells in our data using expression of **EpCAM (Epithelial cell adhesion molecule)** as a marker. Subsequently, we will carry out a DE analysis within the **EpCAM-positive populations**.

Secondly, we will compare the T cell populations of **ICB vs. ICBdT** to determine differences in **T cell phenotypes**.

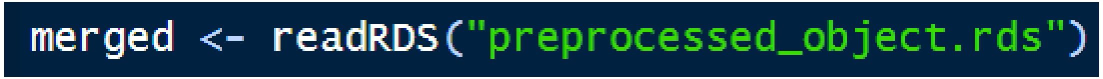
Differential expression analysis

First, let's load the libraries we need:



```
library(Seurat)
library(dplyr)
library(EnhancedVolcano)
library(presto)
```

Then, read in the previously saved Seurat object:

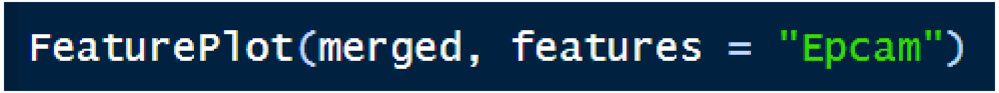


```
merged <- readRDS("preprocessed_object.rds")
```

Gene expression analysis for epithelial cells

We can use Seurat's "**FeaturePlot()**" function to colour each cell by its **Epcam** expression on a UMAP.

FeaturePlot() requires at least 2 arguments – the Seurat object, and the "feature" you want to plot (can be a gene, PC scores, any of the metadata columns, etc.).

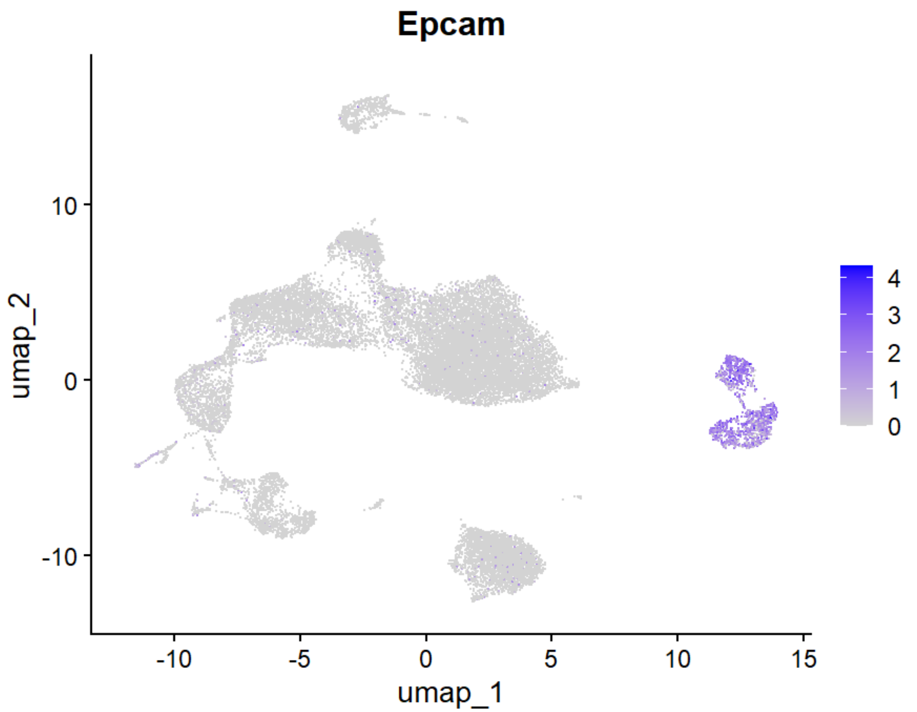


```
FeaturePlot(merged, features = "Epcam")
```

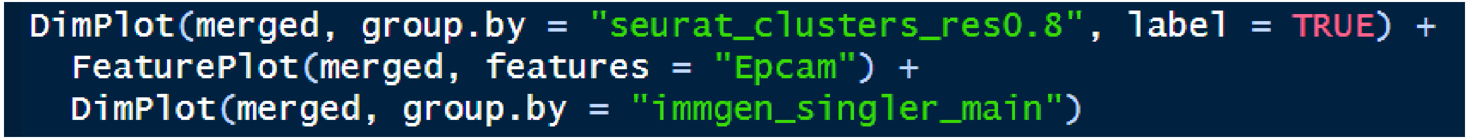
Gene expression analysis for epithelial cells

While there are some Epcam positive cells scattered on the UMAP, there appear to be 2 clusters of cells in the UMAP that we may be able to pull apart as potentially being malignant cells.

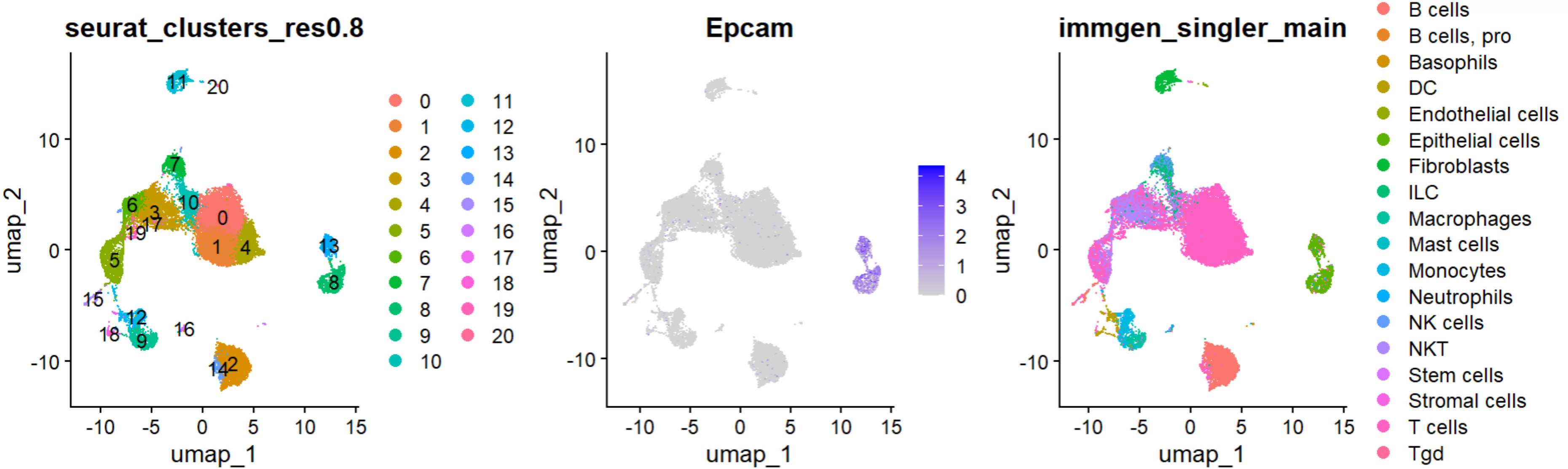
There are a few different ways to identify what these clusters are. We can start by trying to use the DimPlot() along with FeaturePlot().



Gene expression analysis for epithelial cells



```
DimPlot(merged, group.by = "seurat_clusters_res0.8", label = TRUE) +
  FeaturePlot(merged, features = "Epcam") +
  DimPlot(merged, group.by = "immgen_singler_main")
```



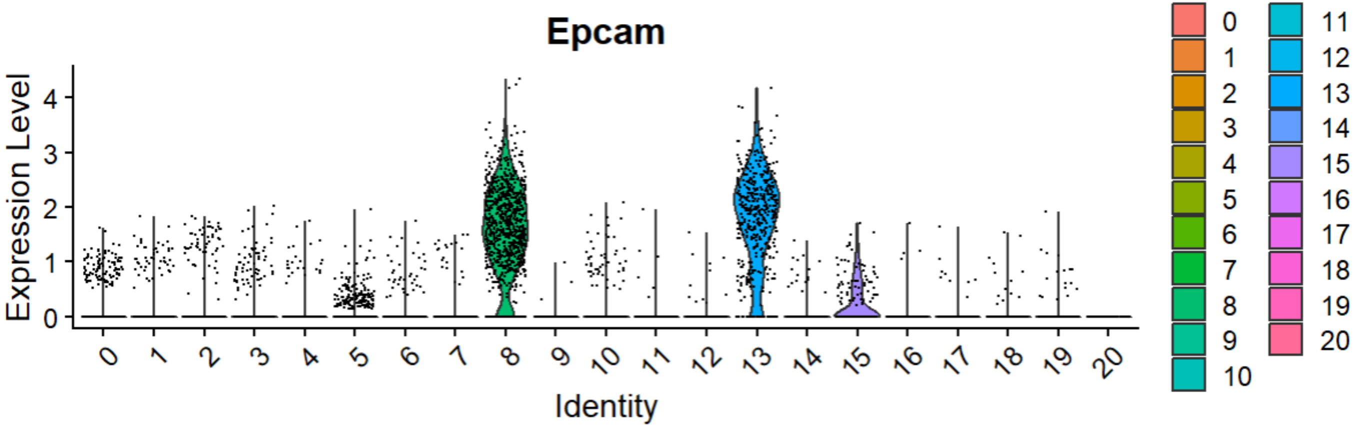
Gene expression analysis for epithelial cells

While the plots generated above make it clear that the clusters of interest are clusters 8 and 13, sometimes it is trickier to determine which cluster we are interested in solely from the UMAP as the clusters may be overlapping.

In this case, a violin plot by "**VlnPlot()**" may be more helpful.

```
VlnPlot(merged, group.by = "seurat_clusters_res0.8", features = "Epcam")
```

Gene expression analysis for epithelial cells



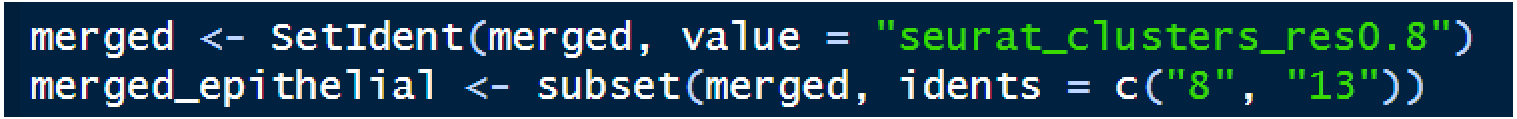
We can confirm that clusters 8 and 13 have the highest expression of Epcam.

However, it is interesting that they are split into 2 clusters. This a good place to use differential expression analysis to determine how these clusters differ from each other.

Differential expression for epithelial cells

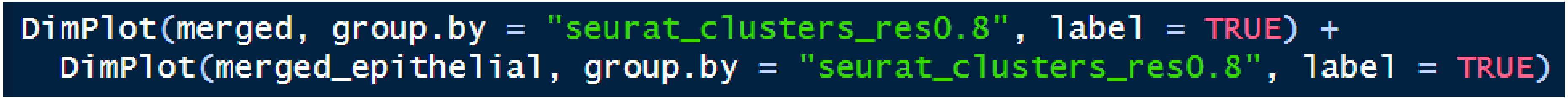
We can begin by restricting the Seurat object to the cells we are interested in.

We will do so using Seurat's `**subset()**` function, it allows us to create an object that is filtered to any values of interest in the metadata column. However, we first need to set the default identity of the Seurat to the metadata column we want to use for the subset, and we can do so using the `**SetIdent()**` function.



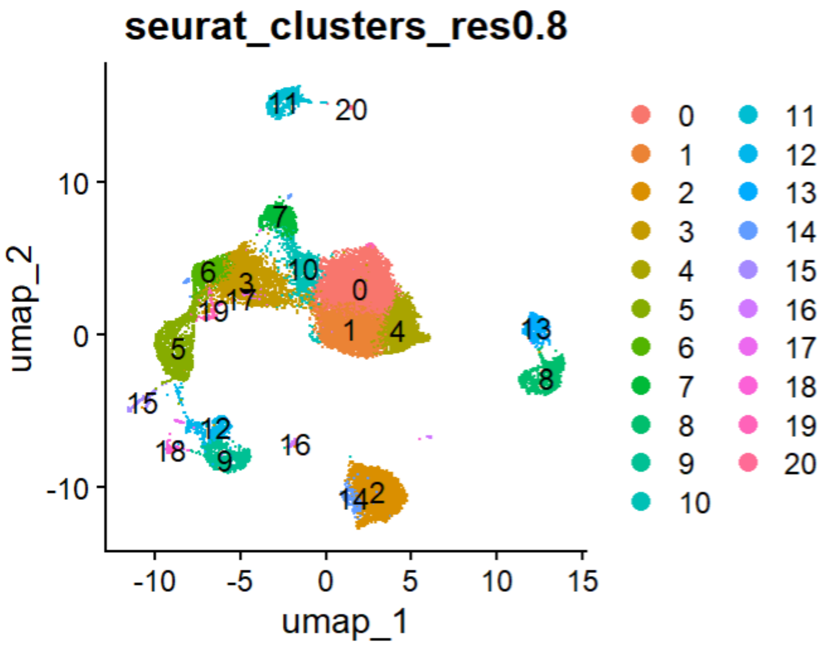
```
merged <- SetIdent(merged, value = "seurat_clusters_res0.8")
merged_epithelial <- subset(merged, idents = c("8", "13"))
```

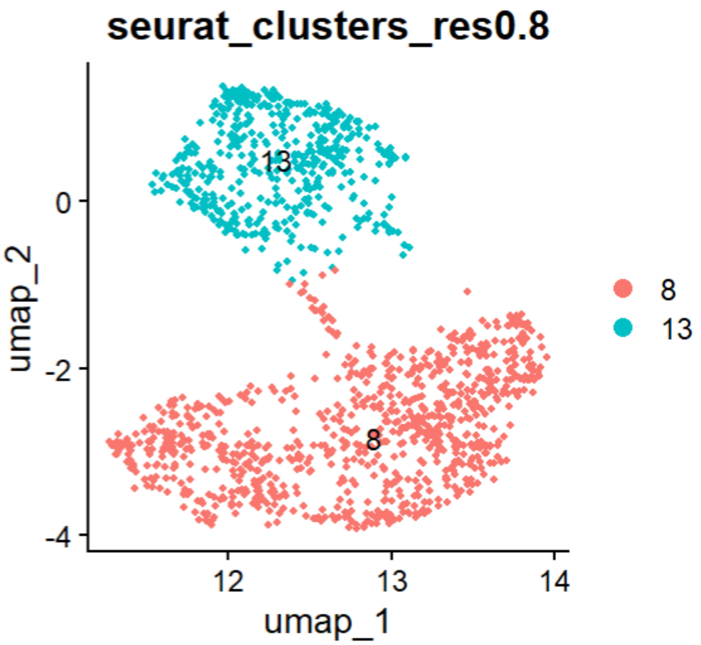
Then, we can plot the original object and the subset object side-by-side to ensure the subset happened as expected.



```
DimPlot(merged, group.by = "seurat_clusters_res0.8", label = TRUE) +
  DimPlot(merged_epithelial, group.by = "seurat_clusters_res0.8", label = TRUE)
```

Differential expression for epithelial cells

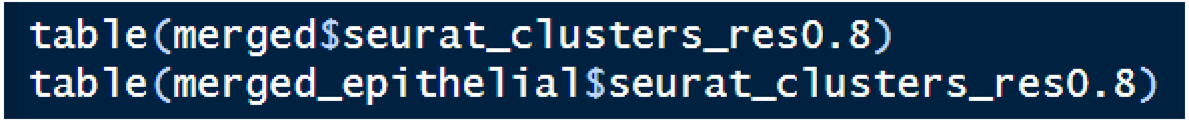




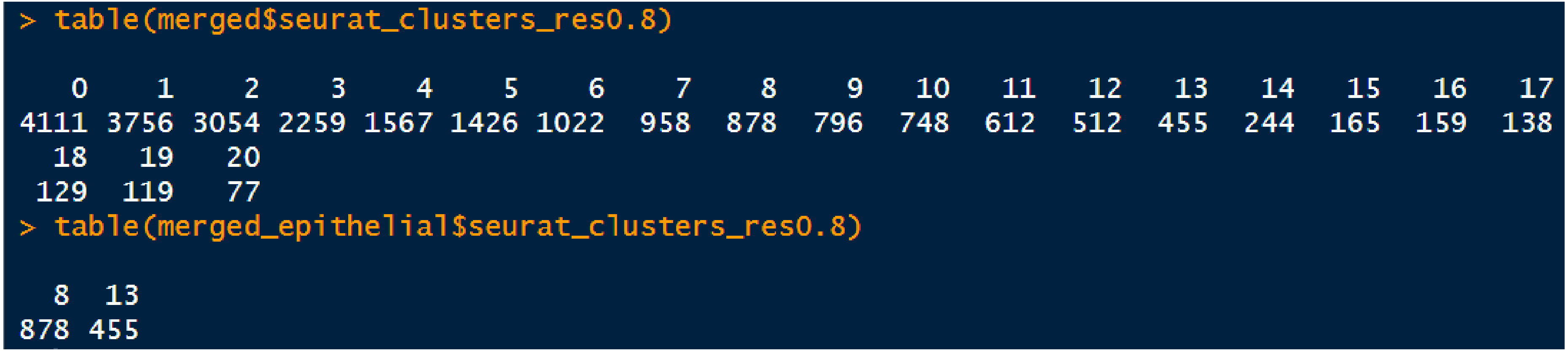
We can see the shape of our clusters are the same between the two plots.

Differential expression for epithelial cells

We can also count the number of cells of each type to confirm they are the same.



```
table(merged$seurat_clusters_res0.8)
table(merged_epithelial$seurat_clusters_res0.8)
```

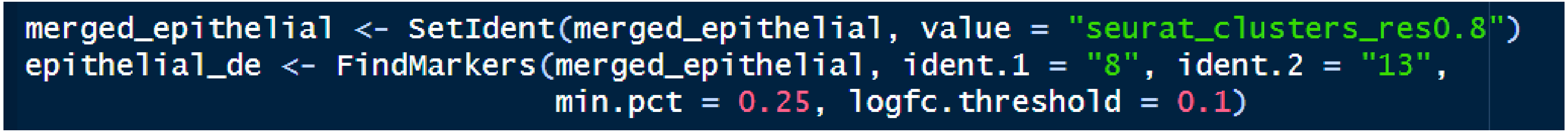


```
> table(merged$seurat_clusters_res0.8)

   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17 
4111 3756 3054 2259 1567 1426 1022  958  878  796  748  612  512  455  244  165  159  138 
  18   19   20 
 129  119   77 
> table(merged_epithelial$seurat_clusters_res0.8)

  8  13 
878 455 
```

Differential expression for epithelial cells

Now we will use Seurat's `**FindMarkers()**` function to carry out a differential expression analysis between both groups.

`FindMarkers()` also requires that we use `SetIdent()` to change the default `**Ident**` to the metadata column we want to use for our comparison.



```
merged_epithelial <- SetIdent(merged_epithelial, value = "seurat_clusters_res0.8")
epithelial_de <- FindMarkers(merged_epithelial, ident.1 = "8", ident.2 = "13",
                             min.pct = 0.25, logfc.threshold = 0.1)
```

For this function, we need to specify the clusters we are comparing. The output is a table with genes that are differentially expressed and its corresponding log2FC value.

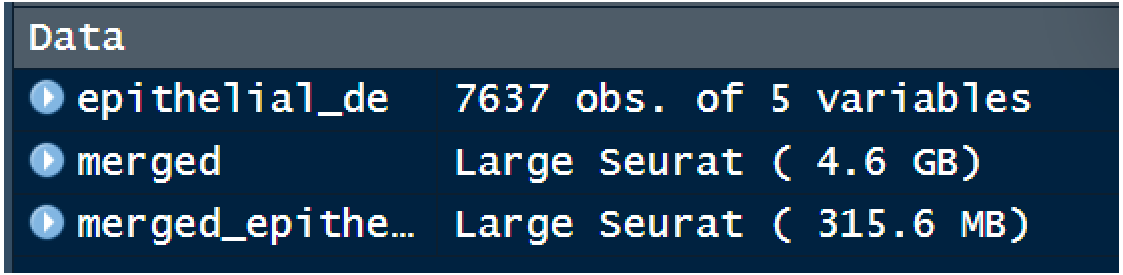
The direction of the log2FC value is of `**ident.1**` with respect to `**ident.2**`. Therefore, genes **upregulated** in ident.1 have **positive value**, downregulated have **negative values**.

Differential expression for epithelial cells

`**min.pct=0.25**`: meaning we only compare genes that are expressed in at least 25% of cells in either cluster.

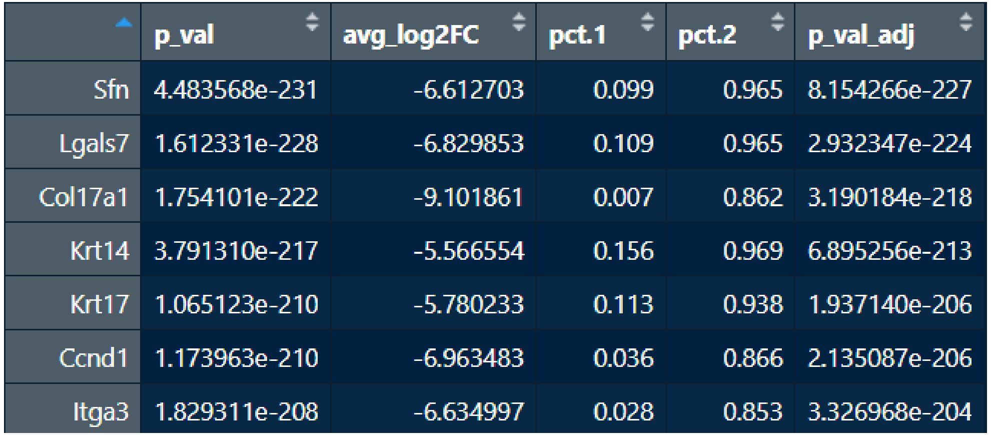
`**logfc.threshold=0.1**`: ensures our results only include genes that have a fold change of less than -0.1 or more than 0.1.

After we get the result, you can click on the object in the top right "Environment" section.



```
Data
epithelial_de     7637 obs. of 5 variables
merged            Large Seurat ( 4.6 GB)
merged_epithe…    Large Seurat ( 315.6 MB)
```

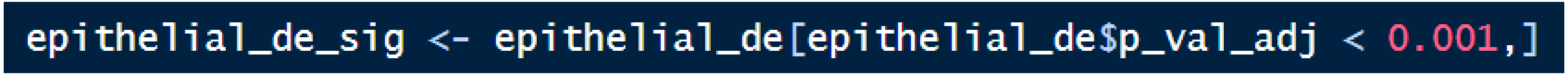
Differential expression for epithelial cells



	p_val	avg_log2FC	pct.1	pct.2	p_val_adj
Sfn	4.483568e-231	-6.612703	0.099	0.965	8.154266e-227
Lgals7	1.612331e-228	-6.829853	0.109	0.965	2.932347e-224
Col17a1	1.754101e-222	-9.101861	0.007	0.862	3.190184e-218
Krt14	3.791310e-217	-5.566554	0.156	0.969	6.895256e-213
Krt17	1.065123e-210	-5.780233	0.113	0.938	1.937140e-206
Ccnd1	1.173963e-210	-6.963483	0.036	0.866	2.135087e-206
Itga3	1.829311e-208	-6.634997	0.028	0.853	3.326968e-204

We get a differentially expressed gene table with 5 columns showing p-values, log2FC, and percentage values in cluster 1 and 2.

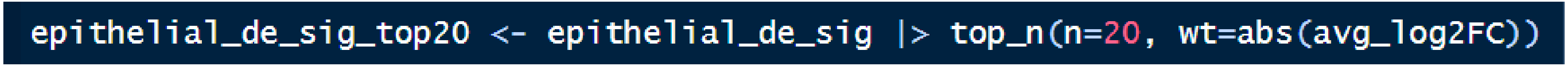
Next, we can further subset this dataframe to only include DE genes that have a significant p-value.



```
epithelial_de_sig <- epithelial_de[epithelial_de$p_val_adj < 0.001,]
```

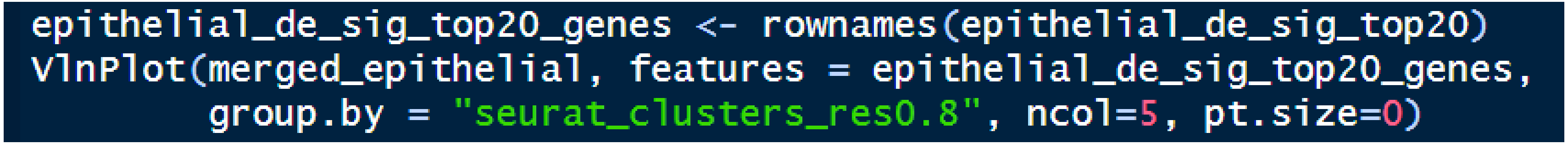
Differential expression for epithelial cells

Then, we further subset the dataframe to top 20 genes that have the highest absolute log2FC value.



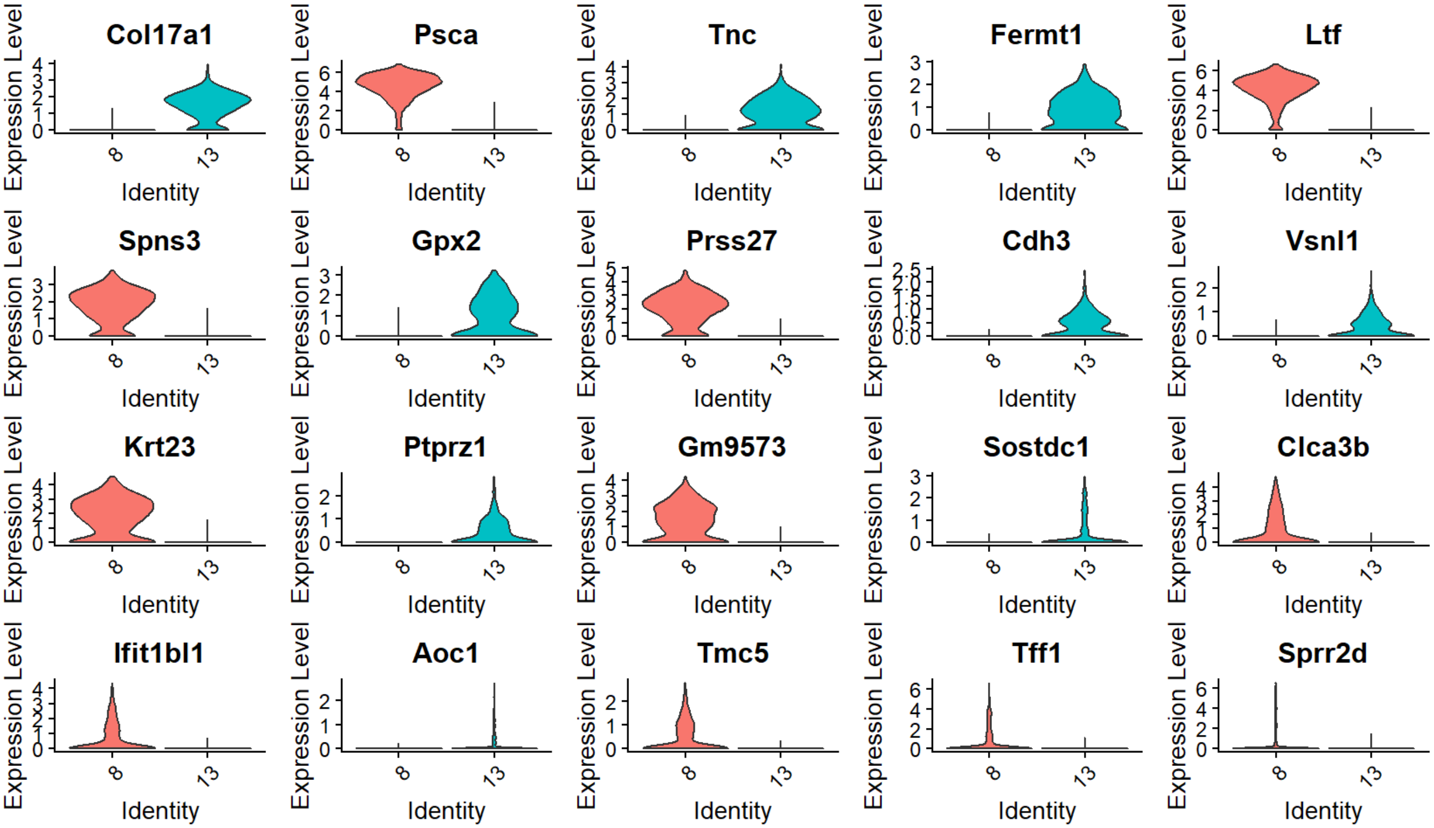
```
epithelial_de_sig_top20 <- epithelial_de_sig |> top_n(n=20, wt=abs(avg_log2FC))
```

There are a few ways we can visualise the differentially expressed genes. We'll start with the Violin and Feature plots from before.



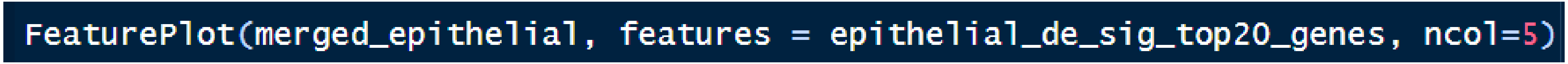
```
epithelial_de_sig_top20_genes <- rownames(epithelial_de_sig_top20)
VlnPlot(merged_epithelial, features = epithelial_de_sig_top20_genes,
        group.by = "seurat_clusters_res0.8", ncol=5, pt.size=0)
```

Differential expression for epithelial cells



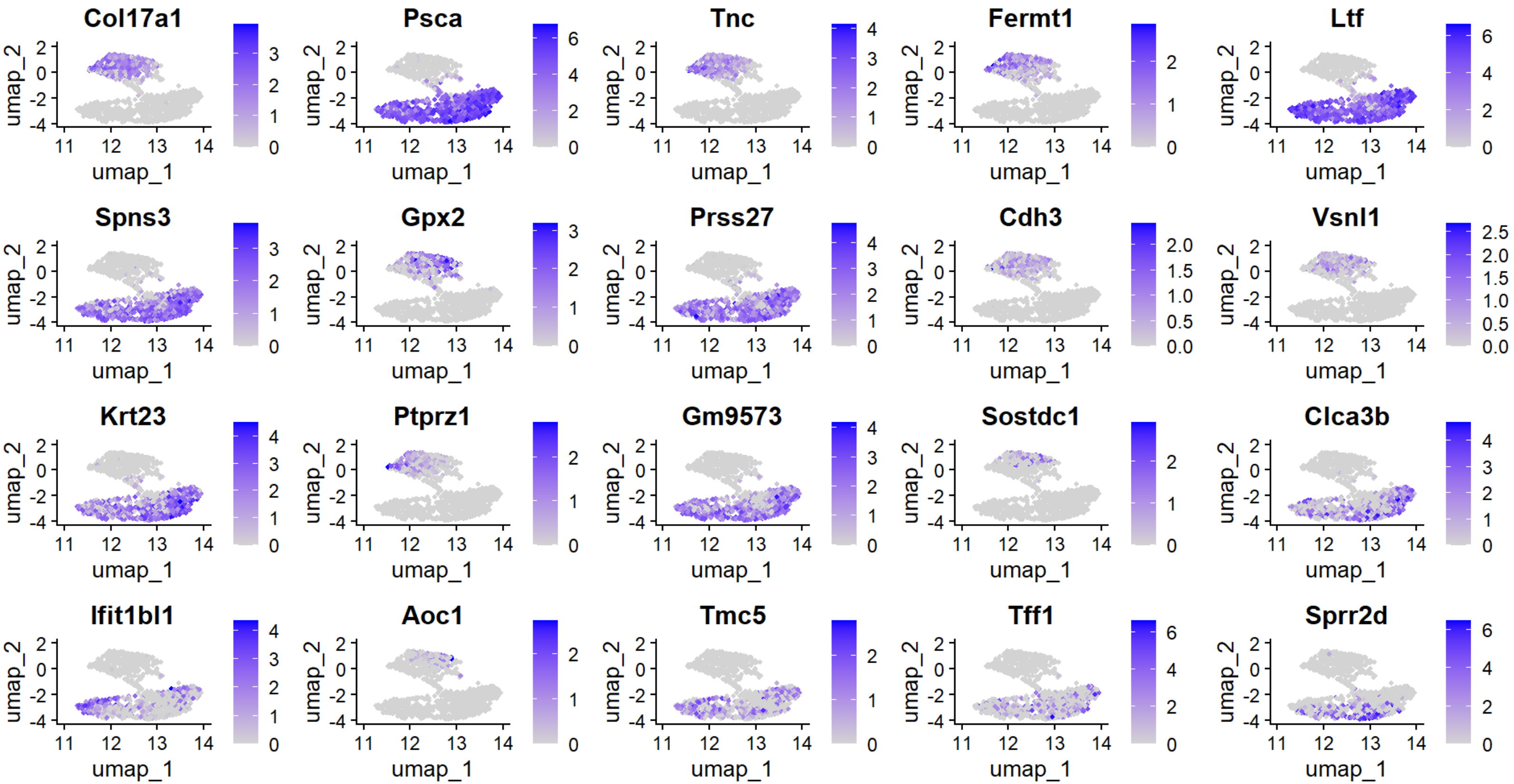
Differential expression for epithelial cells

Plot top 20 genes on UMAP using FeaturePlot().



```
FeaturePlot(merged_epithelial, features = epithelial_de_sig_top20_genes, ncol=5)
```

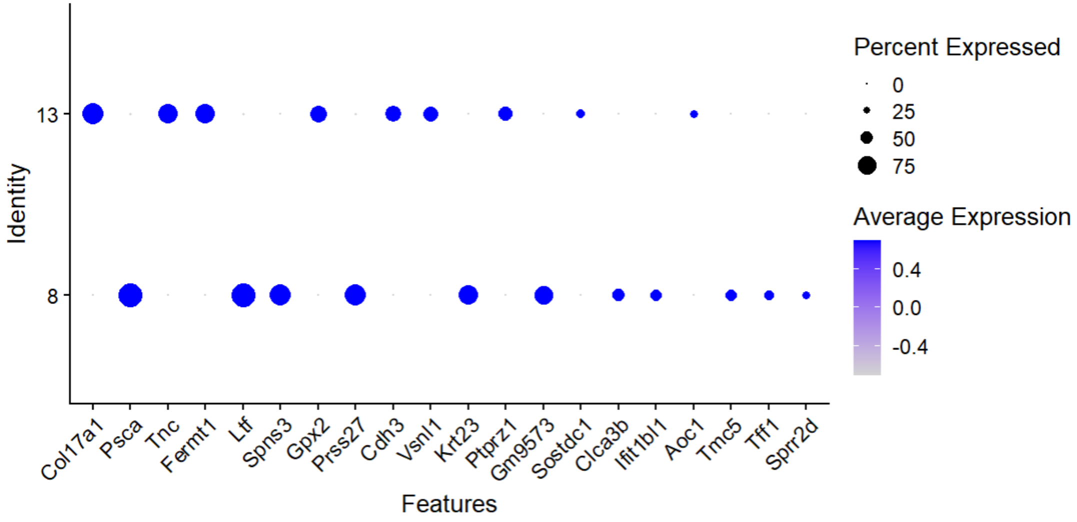
Differential expression for epithelial cells



Differential expression for epithelial cells

We can also visualise DEs using a DotPlot that allows us to capture both the average expression of a gene and the % of cells expressing it.

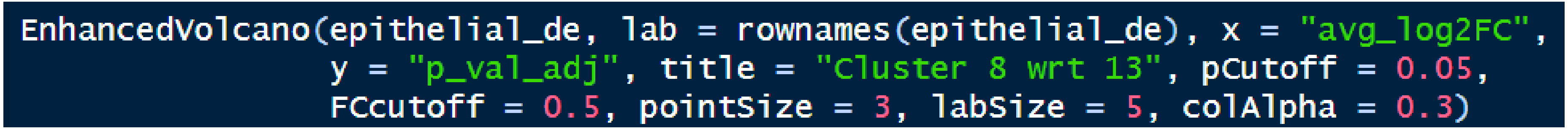
```
DotPlot(merged_epithelial, features = epithelial_de_sig_top20_genes,
        group.by = "seurat_clusters_res0.8") +
  RotatedAxis()
```



Differential expression for epithelial cells

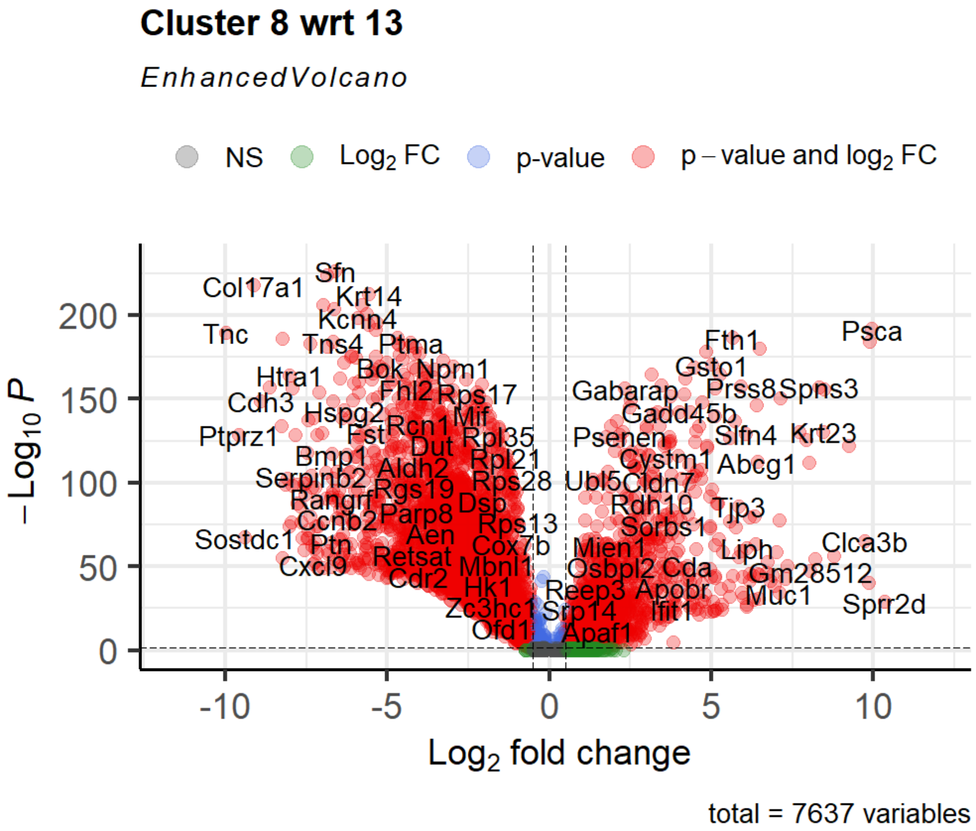
In addition to these built-in Seurat functions, we can also generate a volcano plot using the `**EnhancedVolcano**` package.

For the volcano plot, we can use the unfiltered DE results as the function colours and labels genes based on cutoff values.



```r
EnhancedVolcano(epithelial_de, lab = rownames(epithelial_de), x = "avg_log2FC",
                y = "p_val_adj", title = "Cluster 8 wrt 13", pCutoff = 0.05,
                FCcutoff = 0.5, pointSize = 3, labSize = 5, colAlpha = 0.3)
```

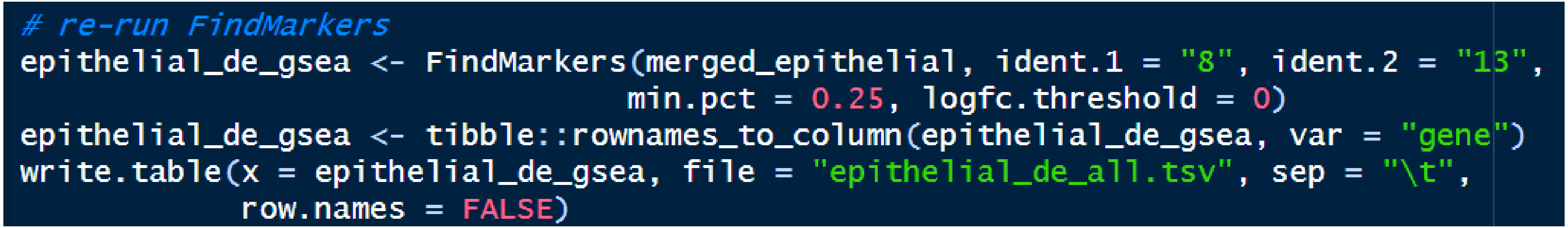
Differential expression for epithelial cells



Differential expression for epithelial cells

To find out how we can figure out what these genes mean, a pathway analysis will shed some light on it.

For now, let's create a TSV file containing our DE result for use later. We will need to re-run `FindMarkers()` with slightly different parameters for this – we will change the `logfc.threshold` parameter to 0, as one of the pathway analysis tool requires all genes to be included.



```
# re-run FindMarkers
epithelial_de_gsea <- FindMarkers(merged_epithelial, ident.1 = "8", ident.2 = "13",
                                 min.pct = 0.25, logfc.threshold = 0)
epithelial_de_gsea <- tibble::rownames_to_column(epithelial_de_gsea, var = "gene")
write.table(x = epithelial_de_gsea, file = "epithelial_de_all.tsv", sep = "\t",
            row.names = FALSE)
```

Pseudo-bulk Differential Expression Analysis

Classic single-cell based differential expression analysis is notorious for having a **high number of false positives**.

There are a few reasons this happens, such as **each cell being considered an independent** observation resulting in **inflated p-values** and it not taking replicates into account.

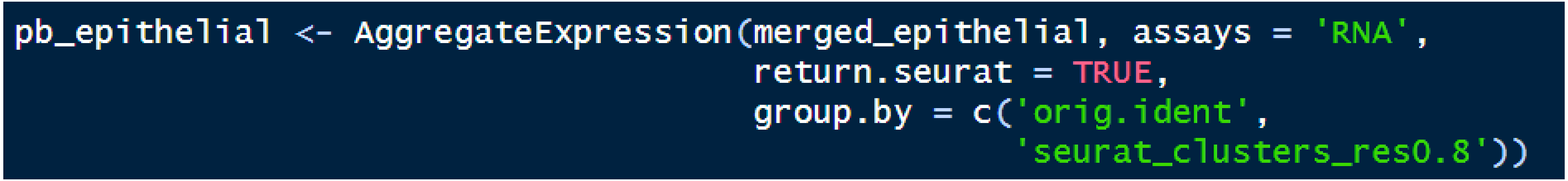
Pseudobulk based differential expression analyses can be incorporated to at least in part overcome this. It is a computational technique that aggregates gene expression data from groups of cells to mimic a bulk RNA-seq experiment.

Instead of analysing each cell individually, pseudobulk analysis sums the raw gene counts for all cells **within a defined group**, from a single biological sample. For example, our interested cluster 8 and 13.

Pseudo-bulk DE as a Solution

Depending on the type of experiment, you're unlikely to always have replicates, but here we do, thus we will carry out a pseudobulk DE analysis and compare the results against our conventional single-cell DE analysis.

The first step is to generate a pseudobulk Seurat object using Seurat's in-built function `` `AggregateExpression()` ``. The function makes groups based on the metadata columns we specify, sums the counts for each gene within the group, and then log normalises and scales the data.

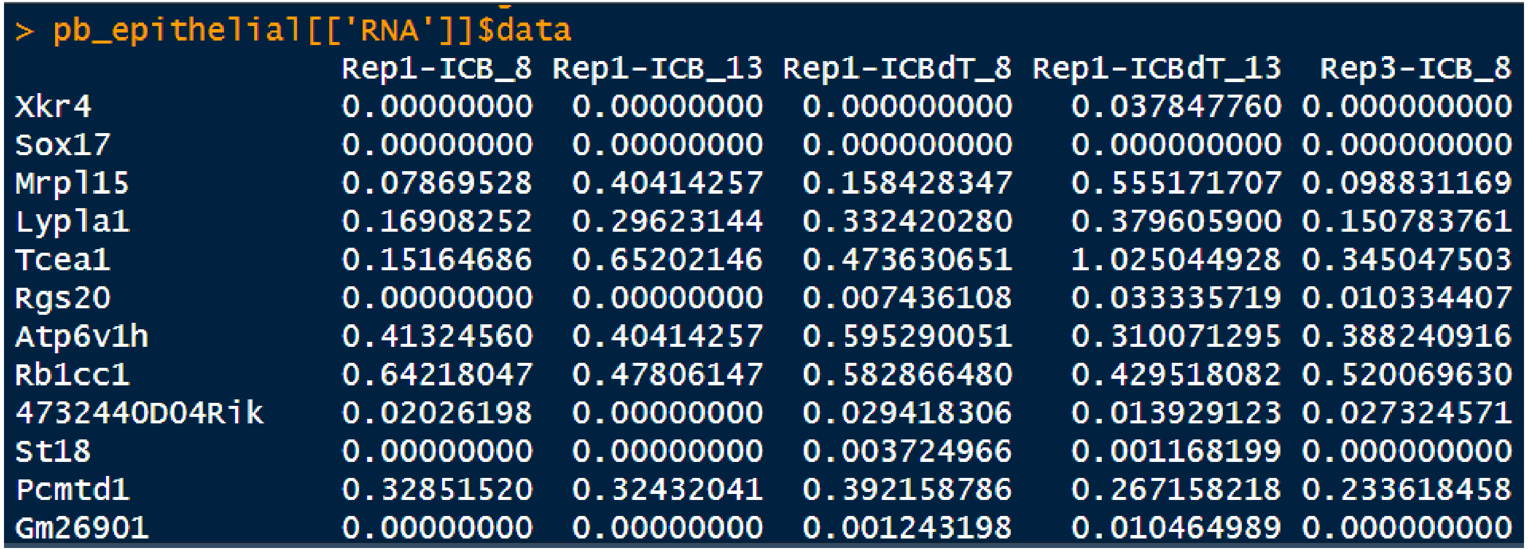


```
pb_epithelial <- AggregateExpression(merged_epithelial, assays = 'RNA',
                                     return.seurat = TRUE,
                                     group.by = c('orig.ident',
                                                  'seurat_clusters_res0.8'))
```

Pseudo-bulk DE as a Solution

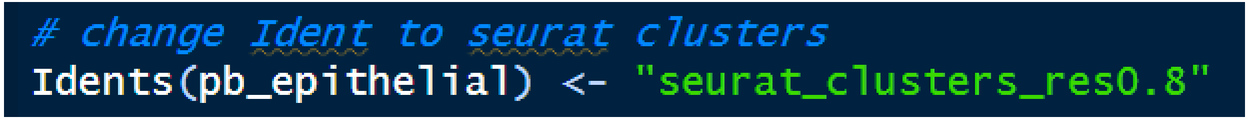
The resulting object is structured like our single-cell Seurat object with **different layers** for the raw counts `` `counts` ``, log normalised `` `data` `` and scaled `` `scale.data` ``.

The aggregated and normalised expression matrix looks like:



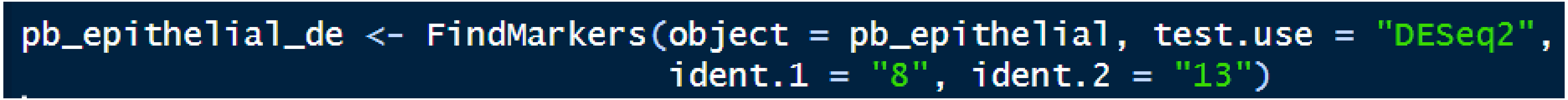
```
> pb_epithelial[['RNA']]$data
              Rep1-ICB_8 Rep1-ICB_13 Rep1-ICBdT_8 Rep1-ICBdT_13  Rep3-ICB_8
Xkr4          0.00000000  0.00000000  0.000000000   0.037847760 0.000000000
Sox17         0.00000000  0.00000000  0.000000000   0.000000000 0.000000000
Mrpl15        0.07869528  0.40414257  0.158428347   0.555171707 0.098831169
Lypla1        0.16908252  0.29623144  0.332420280   0.379605900 0.150783761
Tcea1         0.15164686  0.65202146  0.473630651   1.025044928 0.345047503
Rgs20         0.00000000  0.00000000  0.007436108   0.033335719 0.010334407
Atp6v1h       0.41324560  0.40414257  0.595290051   0.310071295 0.388240916
Rb1cc1        0.64218047  0.47806147  0.582866480   0.429518082 0.520069630
4732440D04Rik 0.02026198  0.00000000  0.029418306   0.013929123 0.027324571
St18          0.00000000  0.00000000  0.003724966   0.001168199 0.000000000
Pcmtd1        0.32851520  0.32432041  0.392158786   0.267158218 0.233618458
Gm26901       0.00000000  0.00000000  0.001243198   0.010464989 0.000000000
```

Pseudo-bulk DE as a Solution



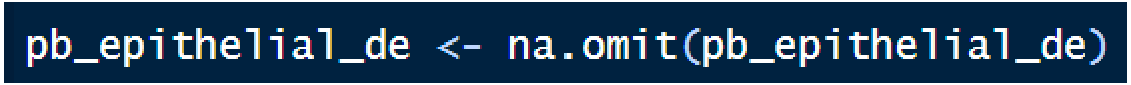
```r
# change Ident to seurat clusters
Idents(pb_epithelial) <- "seurat_clusters_res0.8"
```

Then, we can use sample as our replicates and compare **cluster 8 and 13** for differential gene expression analysis using method `**DESeq2**`.



```r
pb_epithelial_de <- FindMarkers(object = pb_epithelial, test.use = "DESeq2",
                                ident.1 = "8", ident.2 = "13")
```

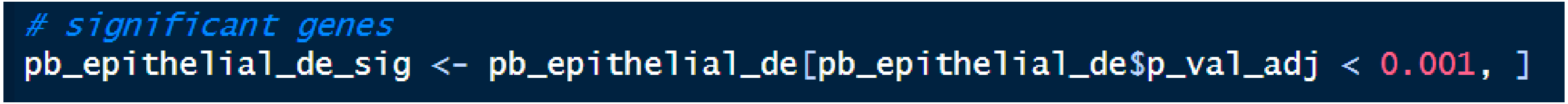
Remove NA values:



```r
pb_epithelial_de <- na.omit(pb_epithelial_de)
```

Pseudo-bulk DE as a Solution

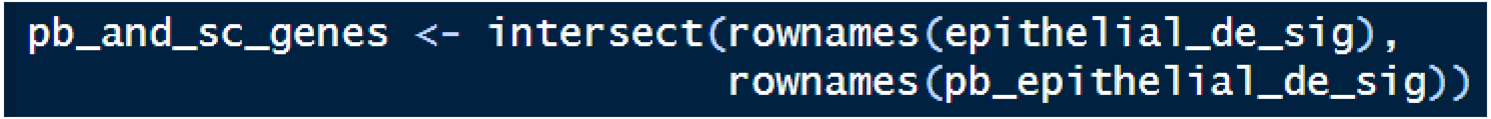
Filter differentially expressed genes with an **adjusted p-value < 0.001**:



```
# significant genes
pb_epithelial_de_sig <- pb_epithelial_de[pb_epithelial_de$p_val_adj < 0.001, ]
```

Then we can compare the identified significant genes between single-cell analysis and pseudobulk analysis.

Genes that present **both in single-cell and pseudobulk** DE:

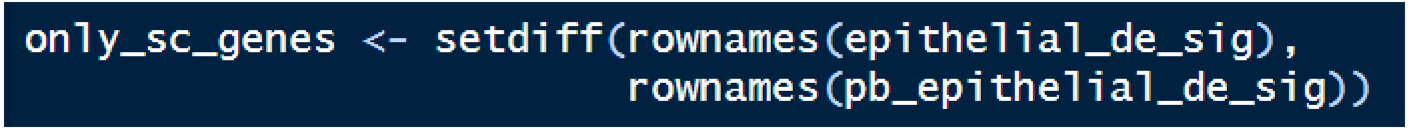


```
pb_and_sc_genes <- intersect(rownames(epithelial_de_sig),
                             rownames(pb_epithelial_de_sig))
```

There are 2536 genes.

Pseudo-bulk DE as a Solution

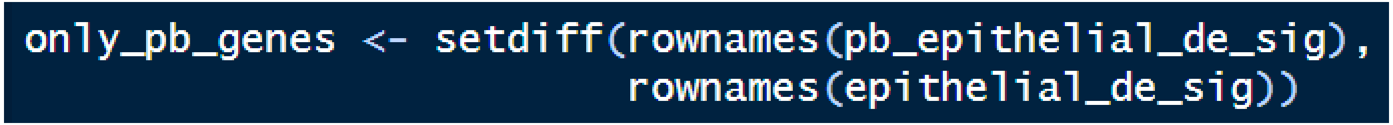
Genes only significant DE in **single-cell**:



```
only_sc_genes <- setdiff(rownames(epithelial_de_sig),
                         rownames(pb_epithelial_de_sig))
```

There are 3333 of them.

Genes only significant DE in **pseudobulk**:



```
only_pb_genes <- setdiff(rownames(pb_epithelial_de_sig),
                         rownames(epithelial_de_sig))
```

There are 434 of them.

Pseudo-bulk DE as a Solution

Genes that were both detected by single-cell and pseudobulk (2786) are most likely to be true positives. Far fewer genes are detected by pseudobulk only (628) compared to single-cell only (3083).

Note: !!!

The pseudobulk genes may also be false positive, as by pseudobulking we lose information about the percent of cells expressing the genes. Therefore, some of these genes may be differentially expressed because they are only expressed in a few cells.

Also, it is unlikely that all ~3000 genes detected only by single-cell are false positives – therefore, it’s difficult to advocate for only one approach, but based on the downstream application one approach may be better than the other.

Pseudo-bulk DE as a Solution

If the goal is to produce a shorter list of genes to follow up on in the wet lab, then the consensus DE gene list may be appropriate.

But if the goal is more exploratory, the single cell genes can be used, but you'll likely want to use **violin plots** and **feature plots** to make sure the genes are indeed differentially expressed.

Like what we did before for the "Epcam" gene.

Pseudo-bulk DE as a Solution

Genes that were both detected by single-cell and pseudobulk (2786) are most likely to be **true positives**. Far fewer genes are detected by pseudobulk only (628) compared to single-cell only (3083).

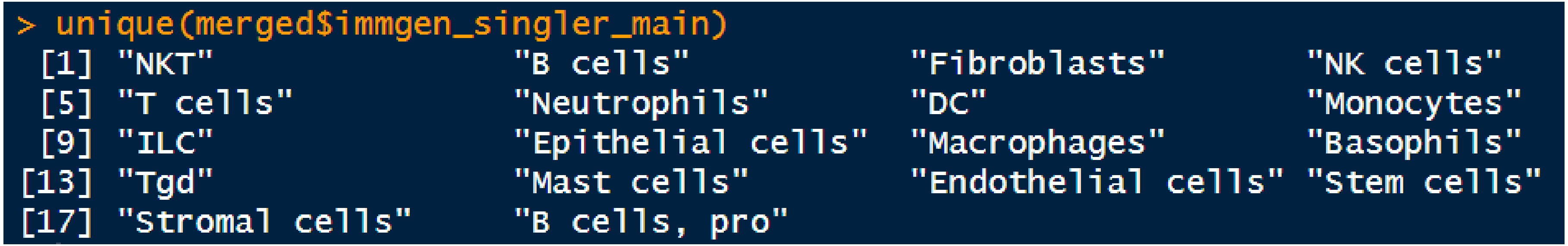
The pseudobulk genes **may also be false positive**, as by pseudobulking we lose information about the percent of cells expressing the genes. Therefore, some of these genes may be differentially expressed because they are only expressed in a few cells.

Also, it is unlikely that all ~3000 genes detected only by single-cell are false positives – therefore, it’s difficult to advocate for only one approach, but based on the downstream application one approach may be better than the other.

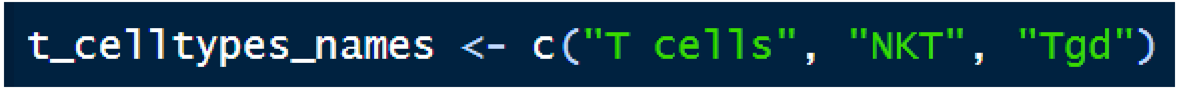
DE Analysis on T cells

For the T cell focused analysis, we will ask how T cells differ **mice ICB vs. ICBdT**. We will start by subsetting our `**merged**` object to only have T cells.

First, let’s check all annotation cell types, then pick those ones related to T cell.



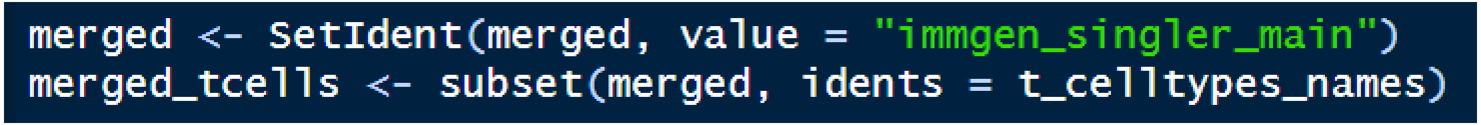
```
> unique(merged$immgen_singler_main)
 [1] "NKT"               "B cells"           "Fibroblasts"       "NK cells"
 [5] "T cells"           "Neutrophils"       "DC"                "Monocytes"
 [9] "ILC"               "Epithelial cells"  "Macrophages"       "Basophils"
[13] "Tgd"               "Mast cells"        "Endothelial cells" "Stem cells"
[17] "Stromal cells"     "B cells, pro"
```



```
t_celltypes_names <- c("T cells", "NKT", "Tgd")
```

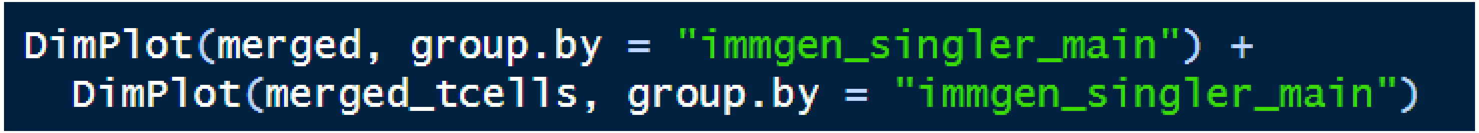
DE Analysis on T cells

Then we can subset the merged Seurat object to only include T cells.



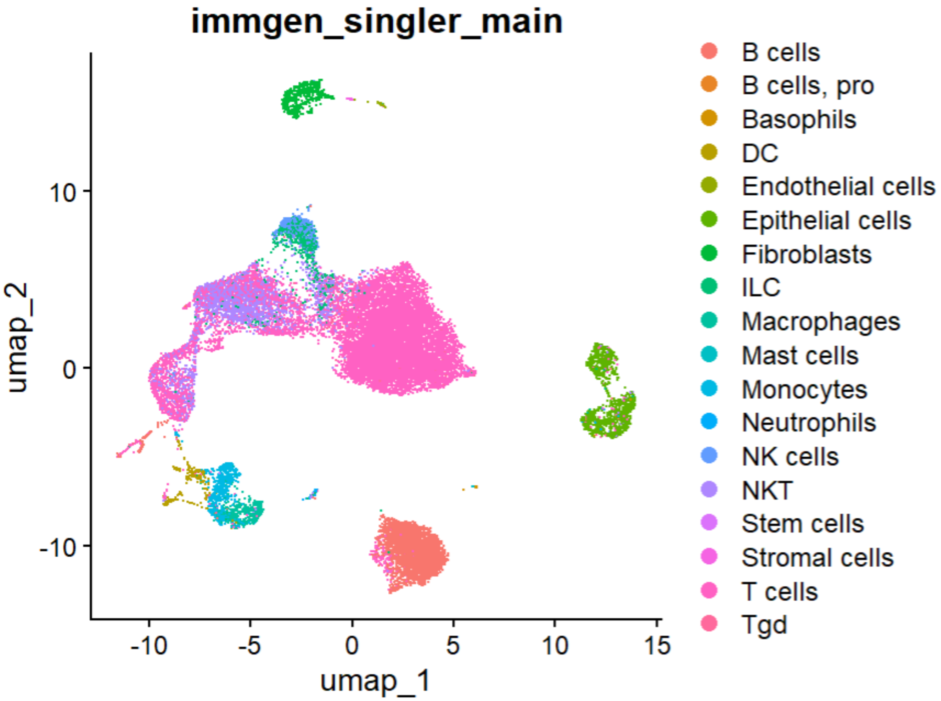
```r
merged <- SetIdent(merged, value = "immgen_singler_main")
merged_tcells <- subset(merged, idents = t_celltypes_names)
```

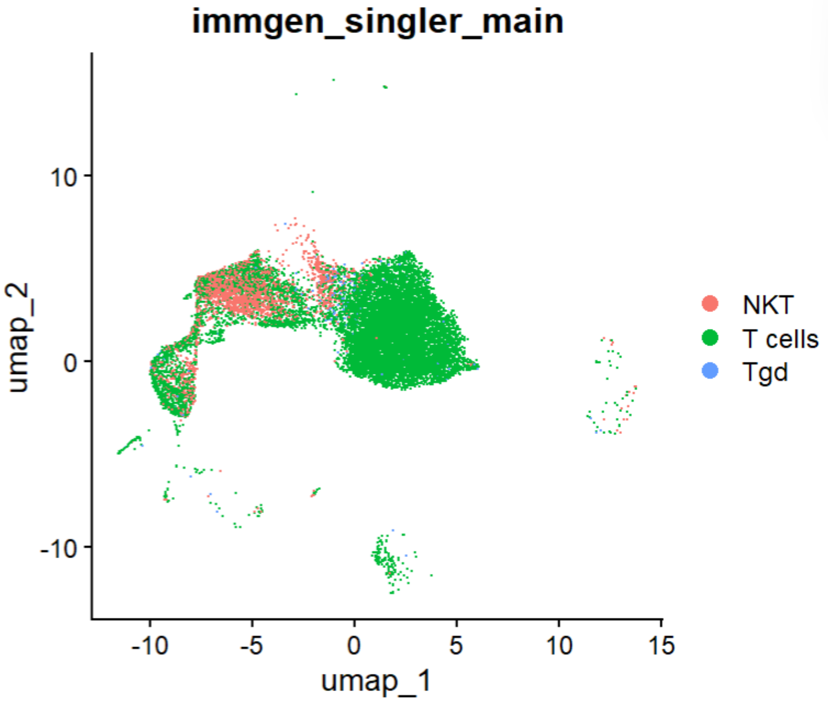
After subsetting, plot the UMAP to confirm we have done it correctly



```r
DimPlot(merged, group.by = "immgen_singler_main") +
  DimPlot(merged_tcells, group.by = "immgen_singler_main")
```

DE Analysis on T cells

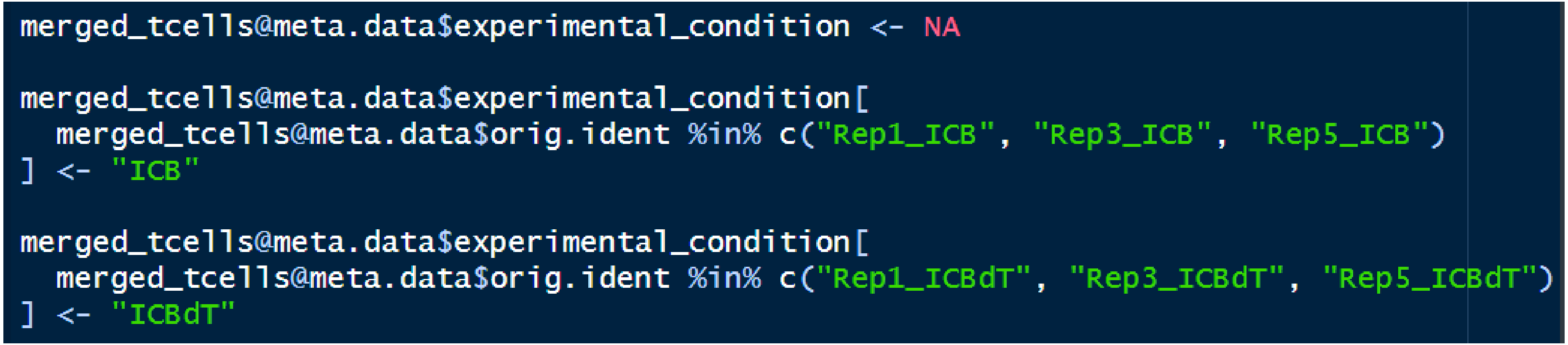




DE Analysis on T cells

Now we want to compare T cells between ICB and ICBdT, we need to know which cell comes from which. In the metadata table of our object, we have a column `orig.ident` which contains information for our 6 samples.

But we want to group each 3 samples together for each condition then compare. We can create a new column to store this condition information.

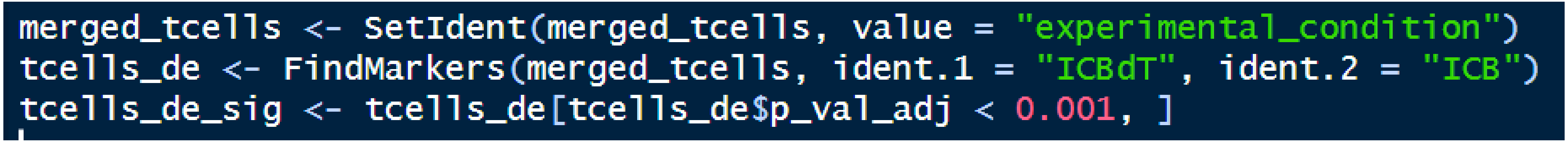


```
merged_tcells@meta.data$experimental_condition <- NA

merged_tcells@meta.data$experimental_condition[
  merged_tcells@meta.data$orig.ident %in% c("Rep1_ICB", "Rep3_ICB", "Rep5_ICB")
] <- "ICB"

merged_tcells@meta.data$experimental_condition[
  merged_tcells@meta.data$orig.ident %in% c("Rep1_ICBdT", "Rep3_ICBdT", "Rep5_ICBdT")
] <- "ICBdT"
```

DE Analysis on T cells

With experimental conditions now defined, we can compare the T cells from both groups.

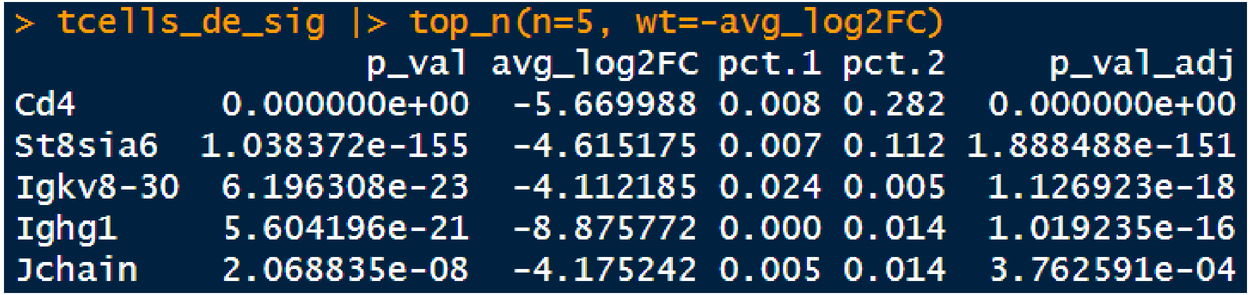
We’ll start by using `**FindMarkers()**` using similar parameters as last time when we comparing cluster 8 and 13. Then select significant genes based adjusted p-values. Then display the top 5 upregulated and downregulated genes.



```
merged_tcells <- SetIdent(merged_tcells, value = "experimental_condition")
tcells_de <- FindMarkers(merged_tcells, ident.1 = "ICBdT", ident.2 = "ICB")
tcells_de_sig <- tcells_de[tcells_de$p_val_adj < 0.001, ]
```

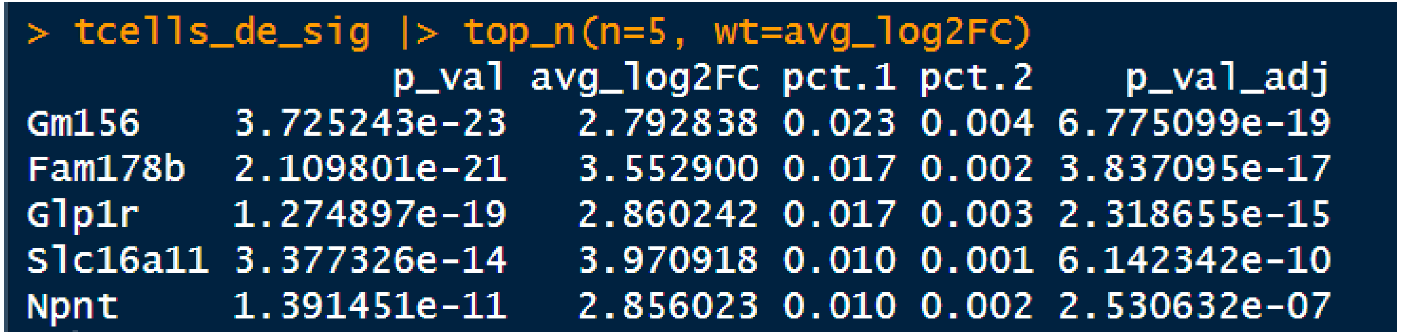
DE Analysis on T cells

Top 5 genes that are **downregulated** in “ICBdT”:



```
> tcells_de_sig |> top_n(n=5, wt=-avg_log2FC)
                 p_val avg_log2FC pct.1 pct.2     p_val_adj
Cd4       0.000000e+00  -5.669988 0.008 0.282  0.000000e+00
St8sia6  1.038372e-155  -4.615175 0.007 0.112 1.888488e-151
Igkv8-30  6.196308e-23  -4.112185 0.024 0.005  1.126923e-18
Ighg1     5.604196e-21  -8.875772 0.000 0.014  1.019235e-16
Jchain    2.068835e-08  -4.175242 0.005 0.014  3.762591e-04
```

Genes are **upregulated** in “ICBdT”:



```
> tcells_de_sig |> top_n(n=5, wt=avg_log2FC)
                p_val avg_log2FC pct.1 pct.2    p_val_adj
Gm156    3.725243e-23   2.792838 0.023 0.004 6.775099e-19
Fam178b  2.109801e-21   3.552900 0.017 0.002 3.837095e-17
Glp1r    1.274897e-19   2.860242 0.017 0.003 2.318655e-15
Slc16a11 3.377326e-14   3.970918 0.010 0.001 6.142342e-10
Npnt     1.391451e-11   2.856023 0.010 0.002 2.530632e-07
```

DE Analysis on T cells

The **most downregulated gene** in ICBdT is **Cd4**. It makes sense because the T cells depletion procedure specifically **targeted CD4 T cells**!!!

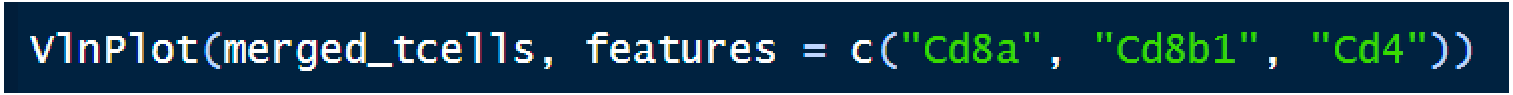
Interestingly, for the list of genes that are **upregulated** in ICBdT, we see **Cd8b1** show up. It could be interesting to see if the CD8 T cells’ phenotype changes based on the treatment.

Let’s subset the object further to only include cells that express CD8 genes.

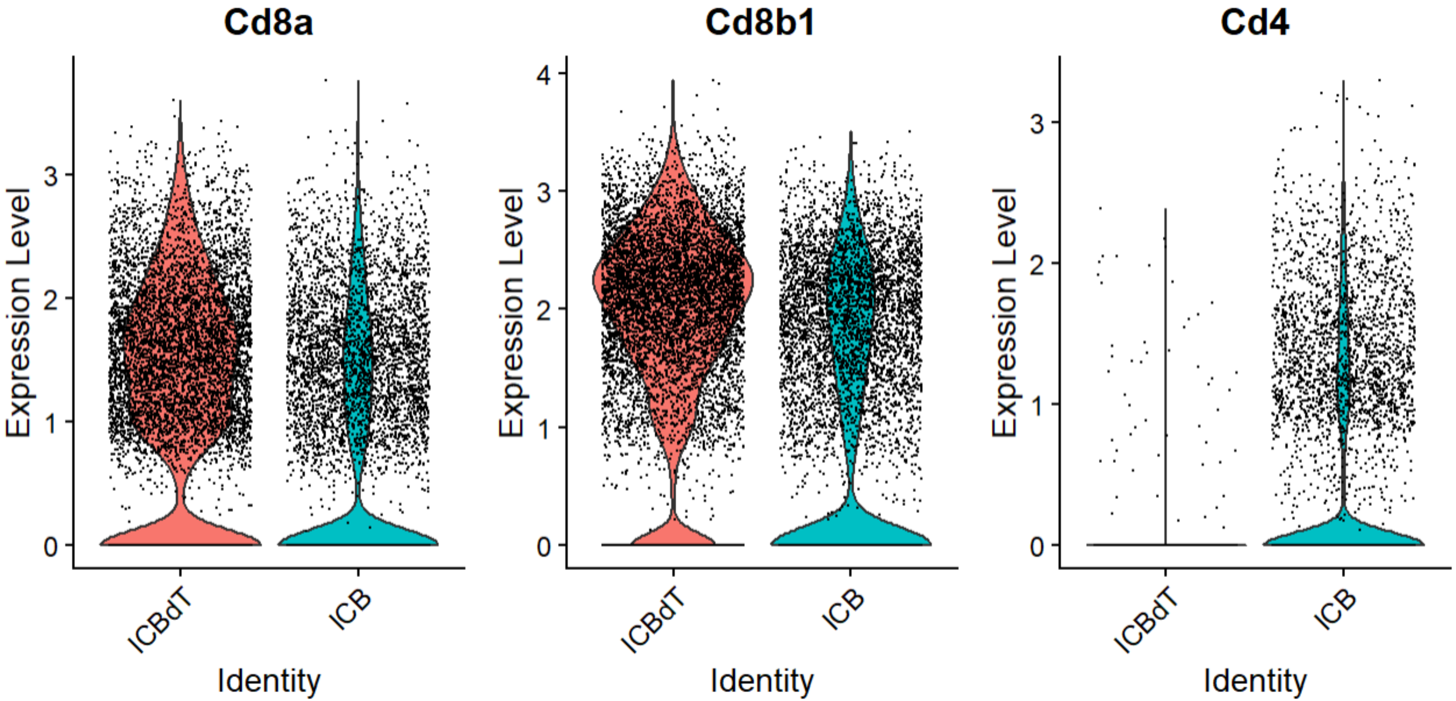
Before we do that, we need to find a suitable threshold to subset. Because in cell type annotation, there isn’t a cell type called CD8, we will have to find CD8 cells manually by comparing the expression of CD8 related genes in each cell.

We can create a **Violin Plot** first to inspect the gene expression pattern for CD8 and CD4 related genes **Cd8a, Cd8b1, Cd4**.

DE Analysis on T cells

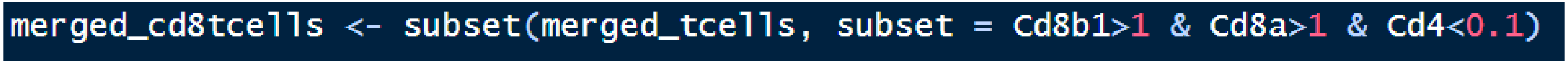
- 

```
VlnPlot(merged_tcells, features = c("Cd8a", "Cd8b1", "Cd4"))
```



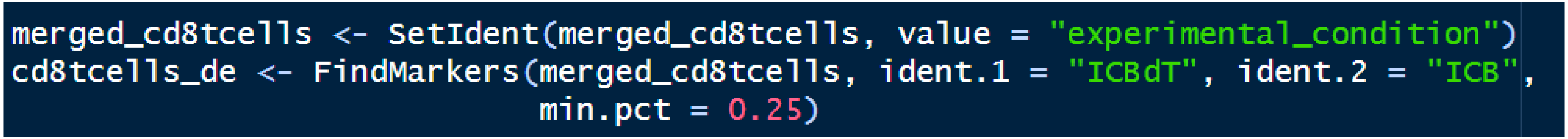
DE Analysis on T cells

We can filter those cells with **CD8 > 1 & CD4 < 0.1**:



```
merged_cd8tcells <- subset(merged_tcells, subset = Cd8b1>1 & Cd8a>1 & Cd4<0.1)
```

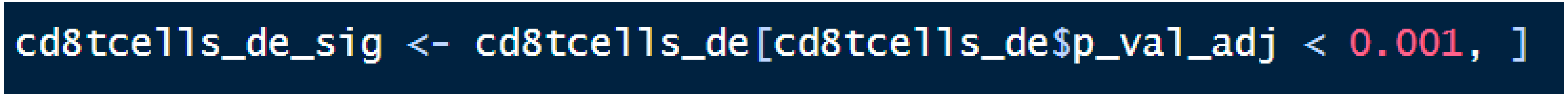
Then, find marker genes:



```
merged_cd8tcells <- SetIdent(merged_cd8tcells, value = "experimental_condition")
cd8tcells_de <- FindMarkers(merged_cd8tcells, ident.1 = "ICBdT", ident.2 = "ICB",
                            min.pct = 0.25)
```

`**min.pct**` only test genes that are detected in more than 25 percent of all cells.

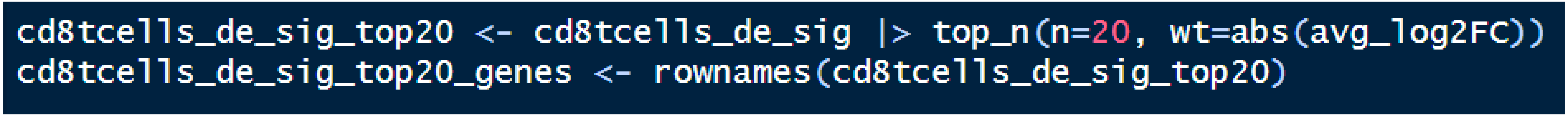
Genes with **adjusted p-value less than 0.001**:



```
cd8tcells_de_sig <- cd8tcells_de[cd8tcells_de$p_val_adj < 0.001, ]
```

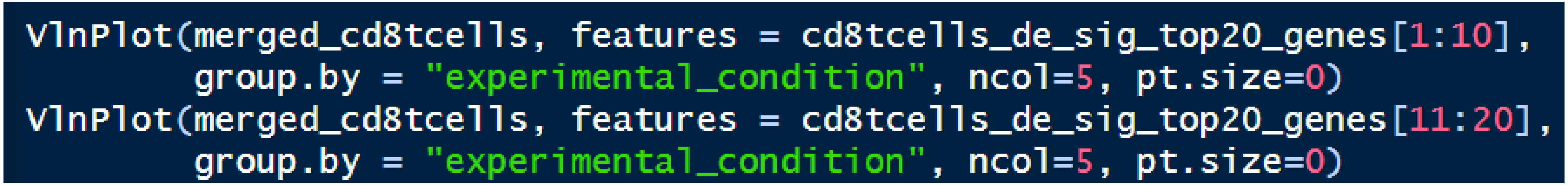
DE Analysis on T cells

Top 20 downregulated and upregulated genes:



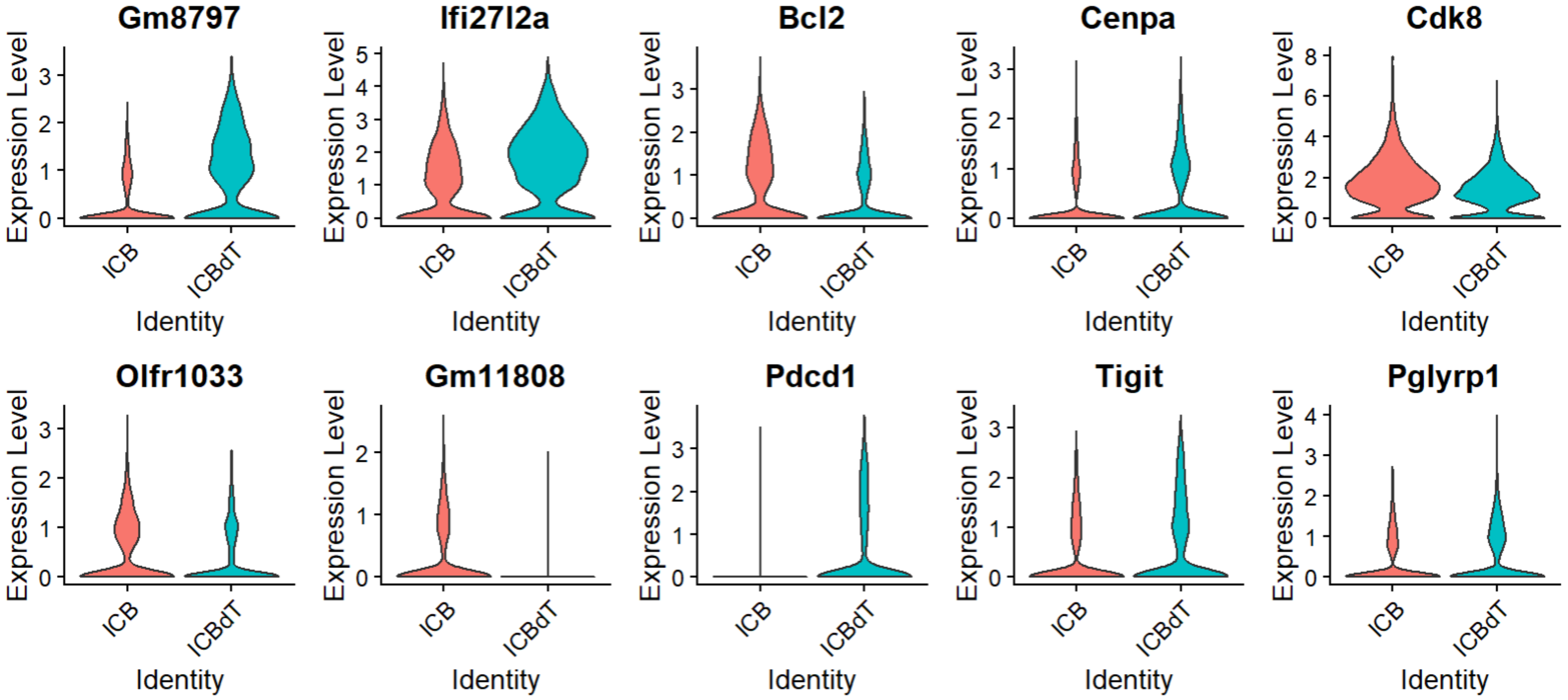
```r
cd8tcells_de_sig_top20 <- cd8tcells_de_sig |> top_n(n=20, wt=abs(avg_log2FC))
cd8tcells_de_sig_top20_genes <- rownames(cd8tcells_de_sig_top20)
```

Compare the expression levels between ICBdT and ICB for the top 20 genes:

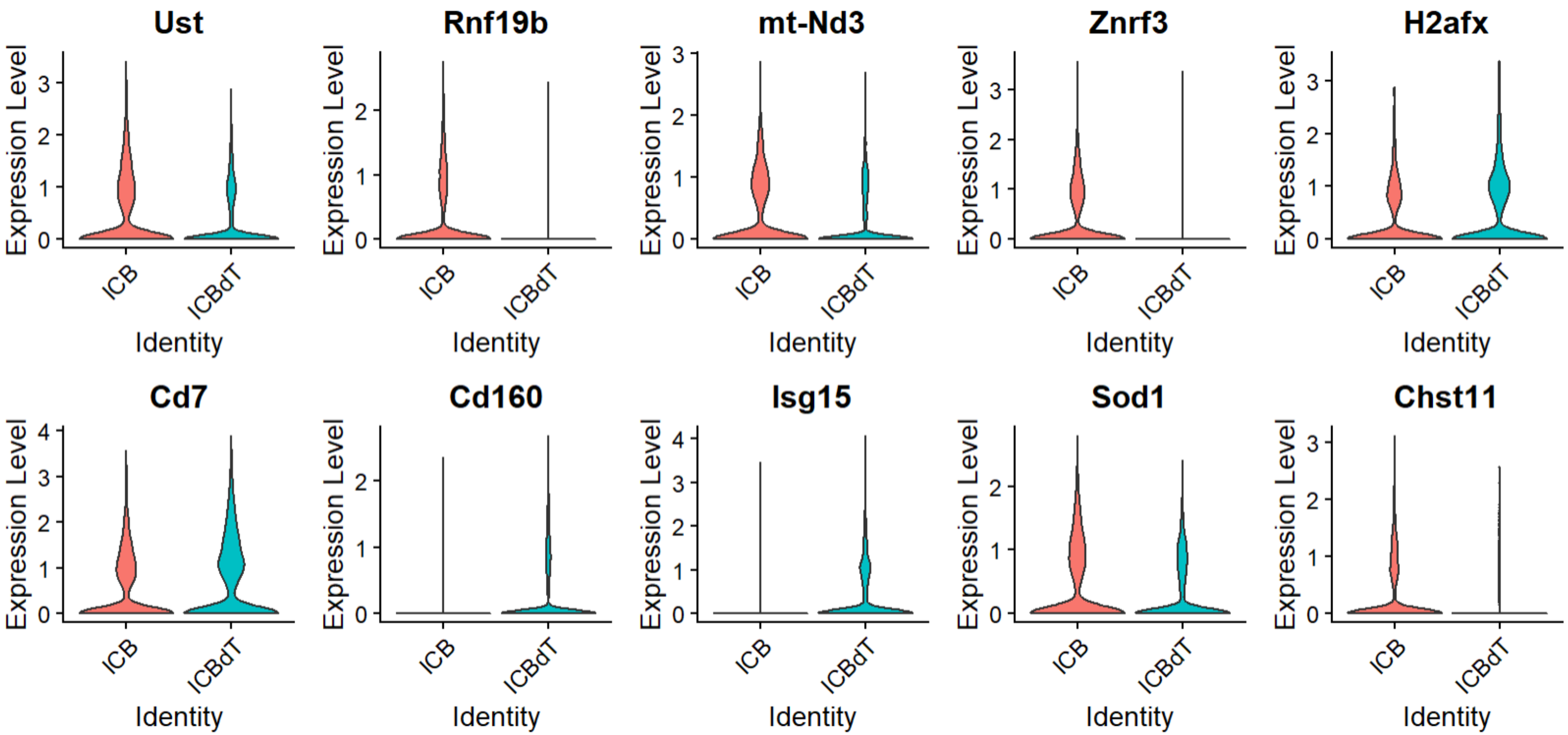


```r
VlnPlot(merged_cd8tcells, features = cd8tcells_de_sig_top20_genes[1:10],
        group.by = "experimental_condition", ncol=5, pt.size=0)
VlnPlot(merged_cd8tcells, features = cd8tcells_de_sig_top20_genes[11:20],
        group.by = "experimental_condition", ncol=5, pt.size=0)
```

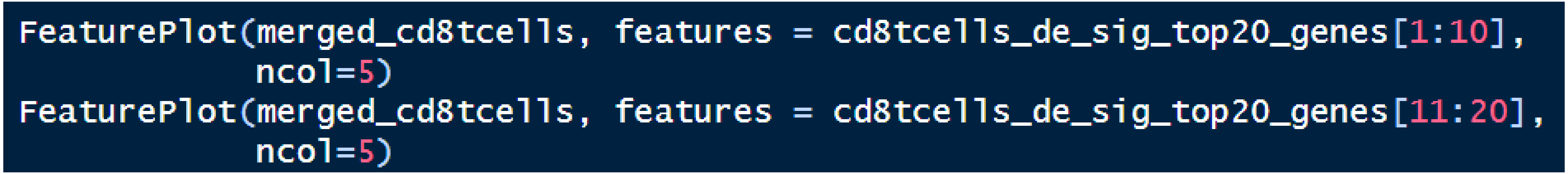
DE Analysis on T cells



DE Analysis on T cells

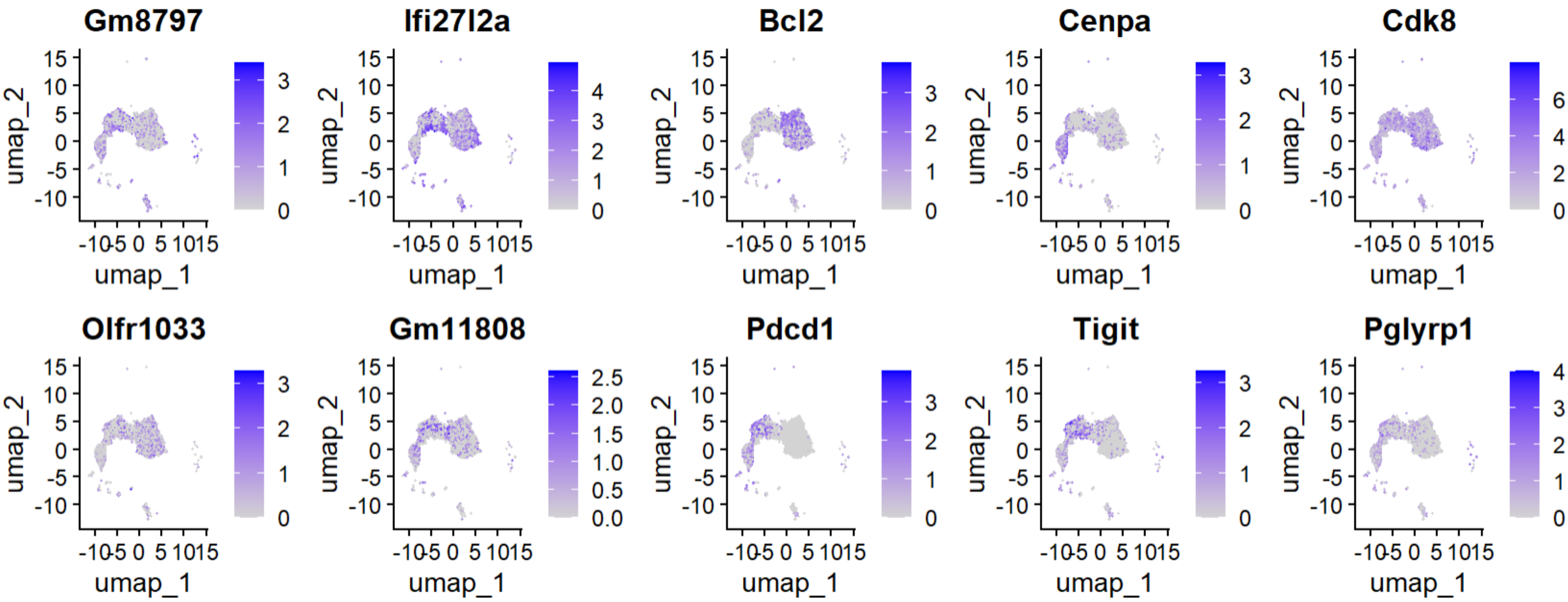


Plot top 20 genes on UMAP



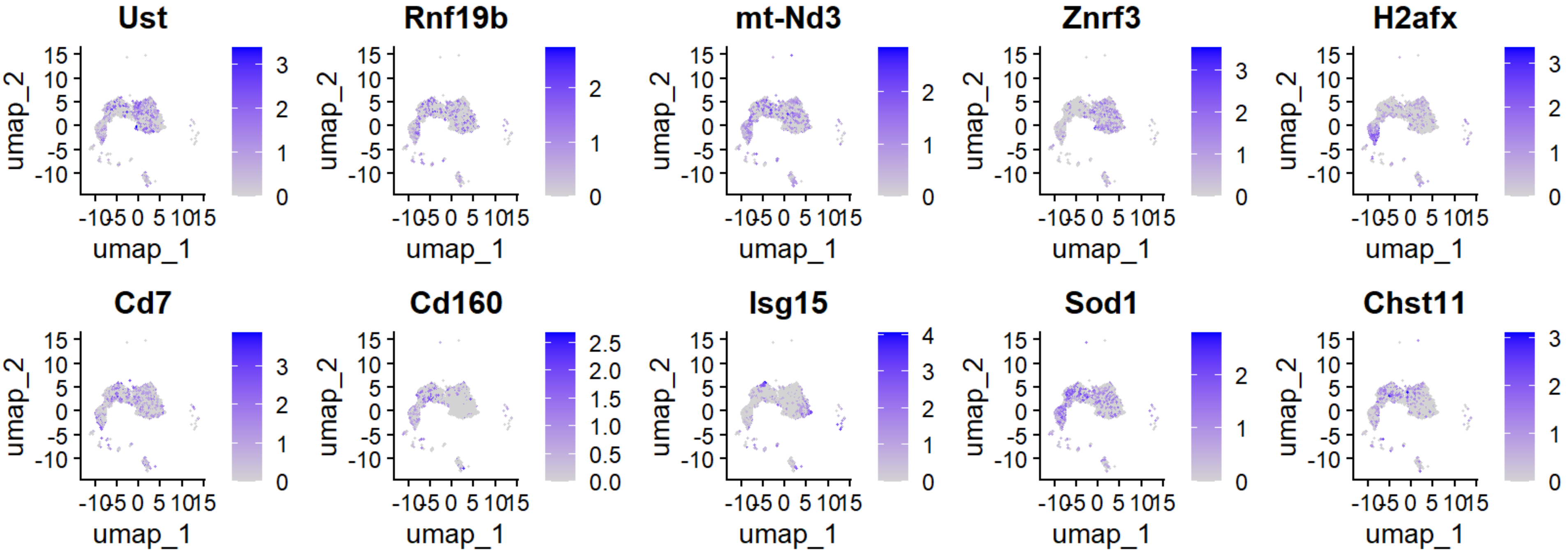
```
FeaturePlot(merged_cd8tcells, features = cd8tcells_de_sig_top20_genes[1:10],
            ncol=5)
FeaturePlot(merged_cd8tcells, features = cd8tcells_de_sig_top20_genes[11:20],
            ncol=5)
```

Plot top 20 genes on UMAP

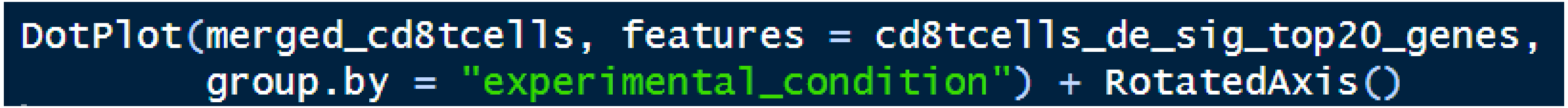


Seurat v5

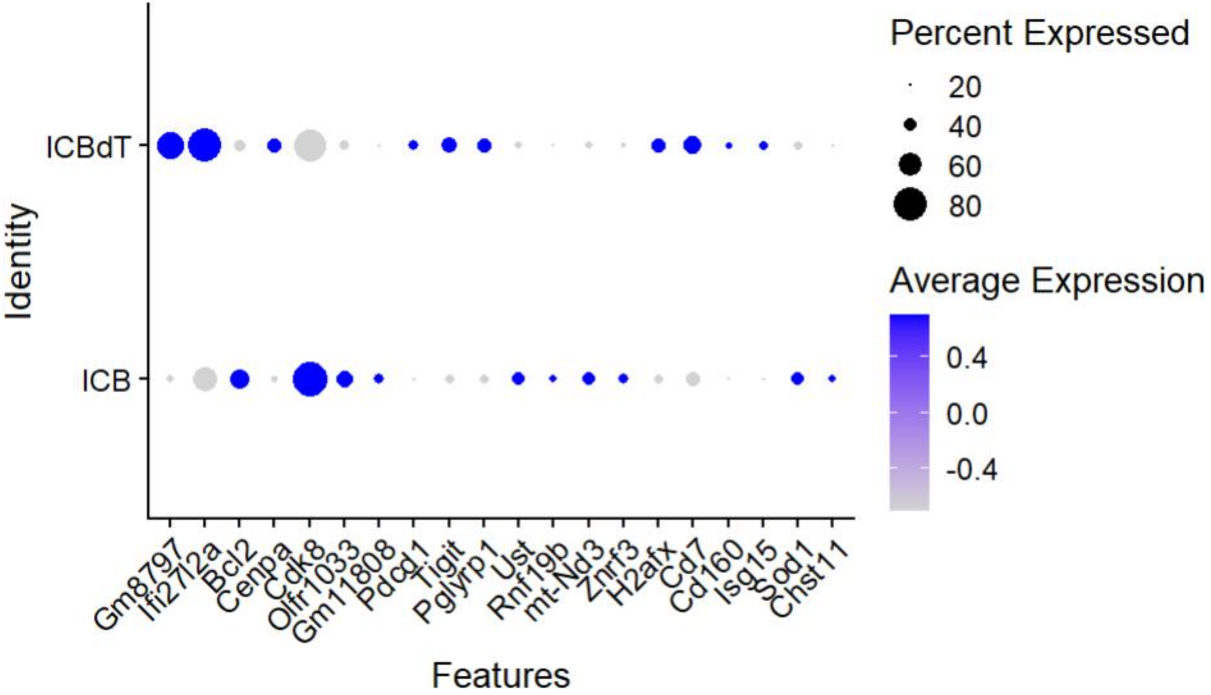
Plot top 20 genes on UMAP



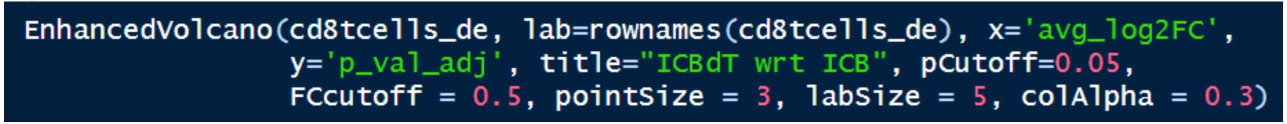
Plot top 20 genes on Dot Plots



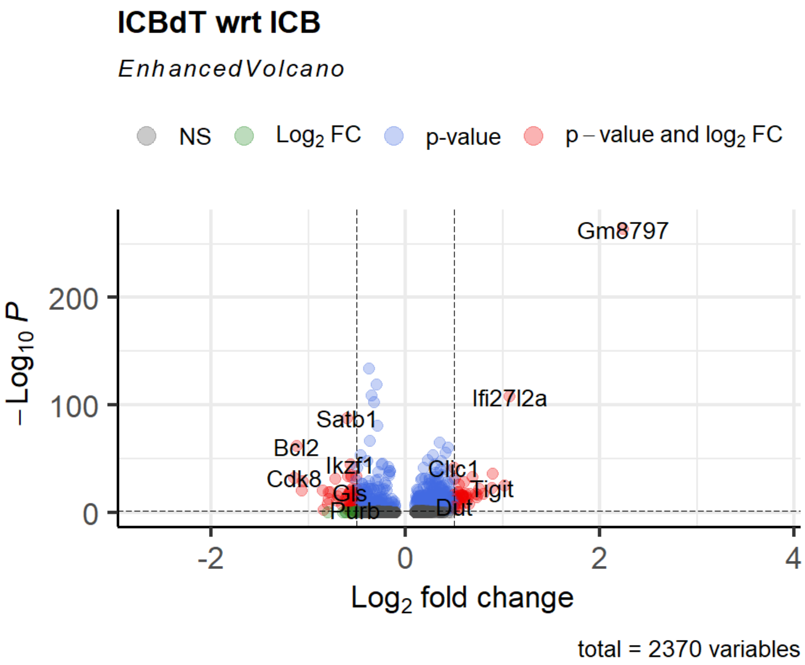
```
DotPlot(merged_cd8tcells, features = cd8tcells_de_sig_top20_genes,
        group.by = "experimental_condition") + RotatedAxis()
```



All DE genes on Volcano Plot



```
EnhancedVolcano(cd8tcells_de, lab=rownames(cd8tcells_de), x='avg_log2FC',
                y='p_val_adj', title="ICBdT wrt ICB", pCutoff=0.05,
                FCcutoff = 0.5, pointSize = 3, labSize = 5, colAlpha = 0.3)
```

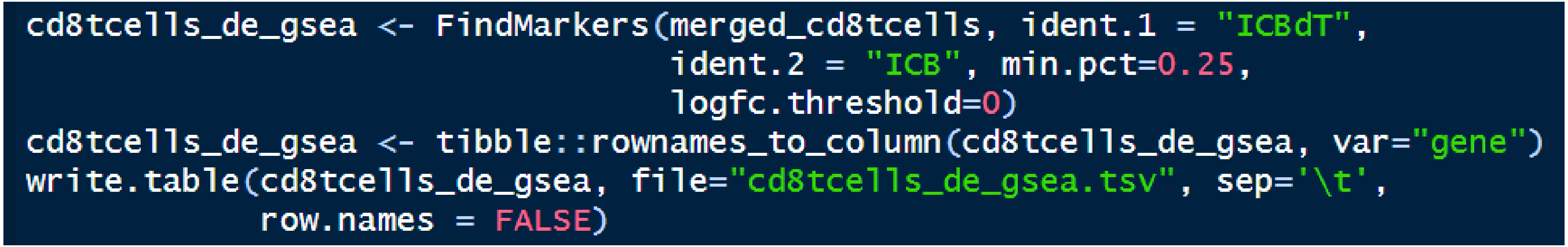


Literature Review for these genes

At this point, you can either start doing literature searches for some of these genes.

We can also use **gene set and pathway analysis** to try and determine what processes the cells may be involved in. And you could also do a Pseudobulk DE Analysis for T cells and CD8 cells to confirm the result.

Now, let’s save this DE result to TSV files for later use:



```
cd8tcells_de_gsea <- FindMarkers(merged_cd8tcells, ident.1 = "ICBdT",
                                 ident.2 = "ICB", min.pct=0.25,
                                 logfc.threshold=0)
cd8tcells_de_gsea <- tibble::rownames_to_column(cd8tcells_de_gsea, var="gene")
write.table(cd8tcells_de_gsea, file="cd8tcells_de_gsea.tsv", sep='\t',
            row.names = FALSE)
```

Thank you

Contact us

Jiajia Li
Research School of Biology

RN Robertson Building, 46 Sullivan's Creek Rd
The Australian National University
Canberra ACT 2600

E jiajia.li1@anu.edu.au

Australian
National
University