

R Notebook

Code ▾

Notebook accompanying the manuscript of Floriddia et. al 2020, Distinct oligodendrocyte populations have spatial preference and different responses to spinal cord injury.

Here we load the data of the first round of single cell sequencing and take samples “GC_7_CC” and “TC_6_CC” to integrate.

Previous runs without integration showed similar results, meaning they are not really experiencing batch effects, but I perform integration anyway to align them as best possible.

anno_10x									
GC_51	GD_52	TC_47	TC_50	GC_32_MC	GC_7_CC	GD_30_CC	GD_31_MC	GD_3_MC	TC_1_MC
1039	2850	903	6192	891	1055	2584	2438	252	TC_6_CC
2266	2037								

Now we perform QC, looking at the percentage of **mitochondrial RNA vs other RNA**, plus other metrics.

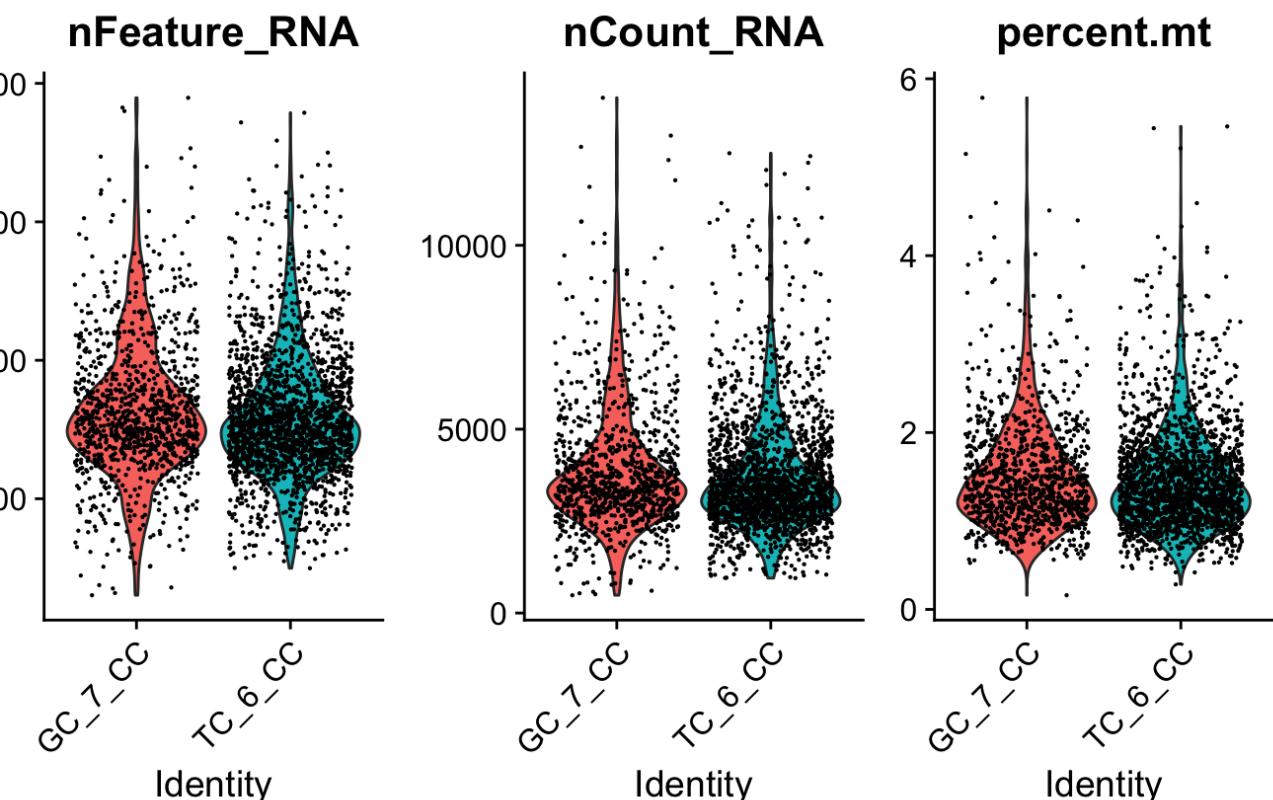
* nFeature_RNA = number of genes

* nCount_RNA = number of UMIs or Counts

* percent.mt = percent of expression of mitochondrial genes versus the rest

Hide

```
# The [[ operator can add columns to object metadata. This is a great place to stash
# QC stats
oligos[["percent.mt"]] <- PercentageFeatureSet(oligos, pattern = "^\$mt-")
# Visualize QC metrics as a violin plot
VlnPlot(oligos, group.by = "Sample", features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3, pt.size = 0.1)
```

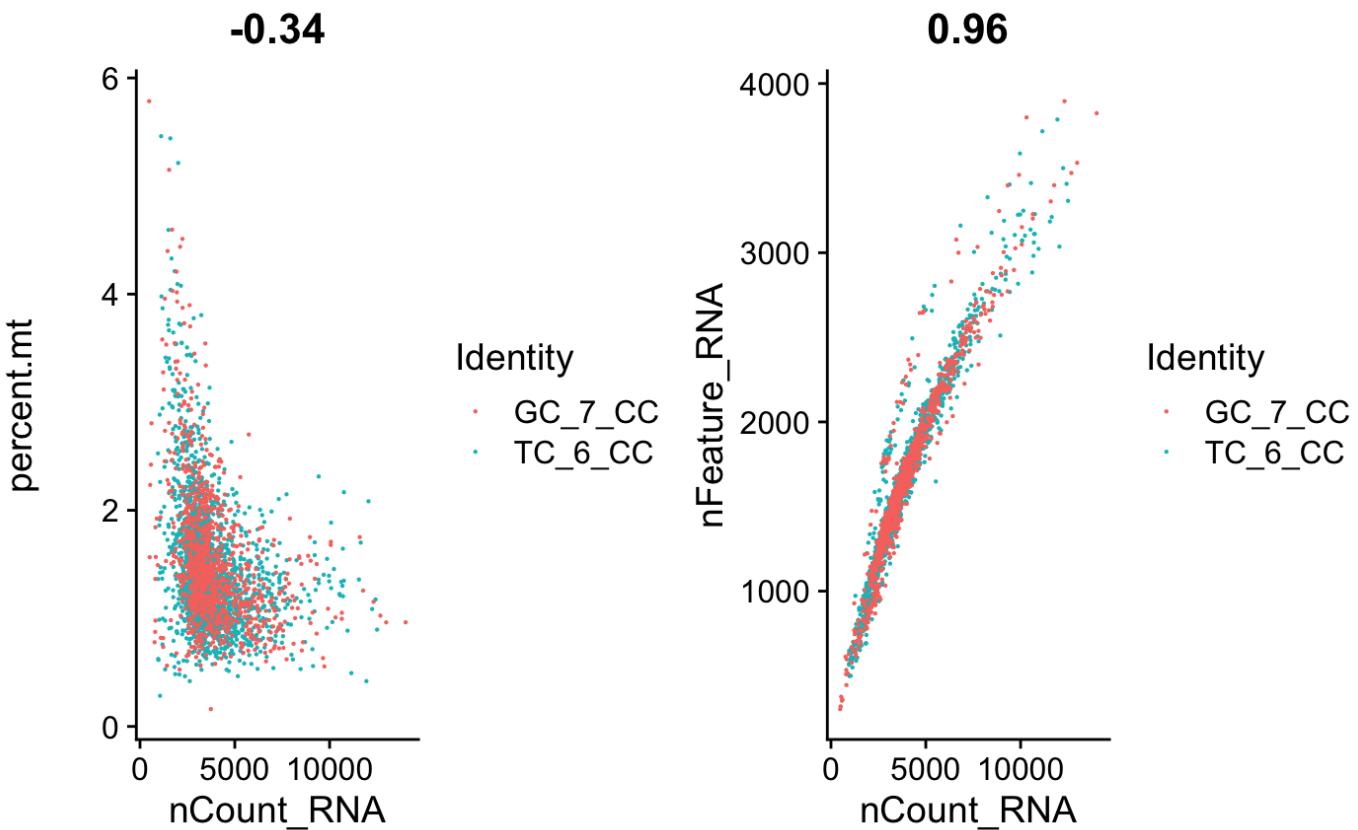


The two samples seem comparable QC-wise, so now we plot the QC information in another way to see if we can estimate thresholds for removing bad cells and perhaps doublets.

Hide

```
# FeatureScatter is typically used to visualize feature-feature relationships, but can be used
# for anything calculated by the object, i.e. columns in object metadata, PC scores etc.
plot1 <- FeatureScatter(oligos, group.by = "Sample", feature1 = "nCount_RNA", feature2 = "percent.mt", pt.size = 0.1)
plot2 <- FeatureScatter(oligos, group.by = "Sample", feature1 = "nCount_RNA", feature2 = "nFeature_RNA", pt.size = 0.1)
CombinePlots(plots = list(plot1, plot2))
```

CombinePlots is being deprecated. Plots should now be combined using the patchwork system.



These samples seem to be performing similarly, which is a great sign for the integration
Now we will remove cells expressing less than 200 genes (to remove bad cells),
and more than 3000 genes (to remove doublets). And remove cells expressing more than 5% mitochondrial
genes.

Hide

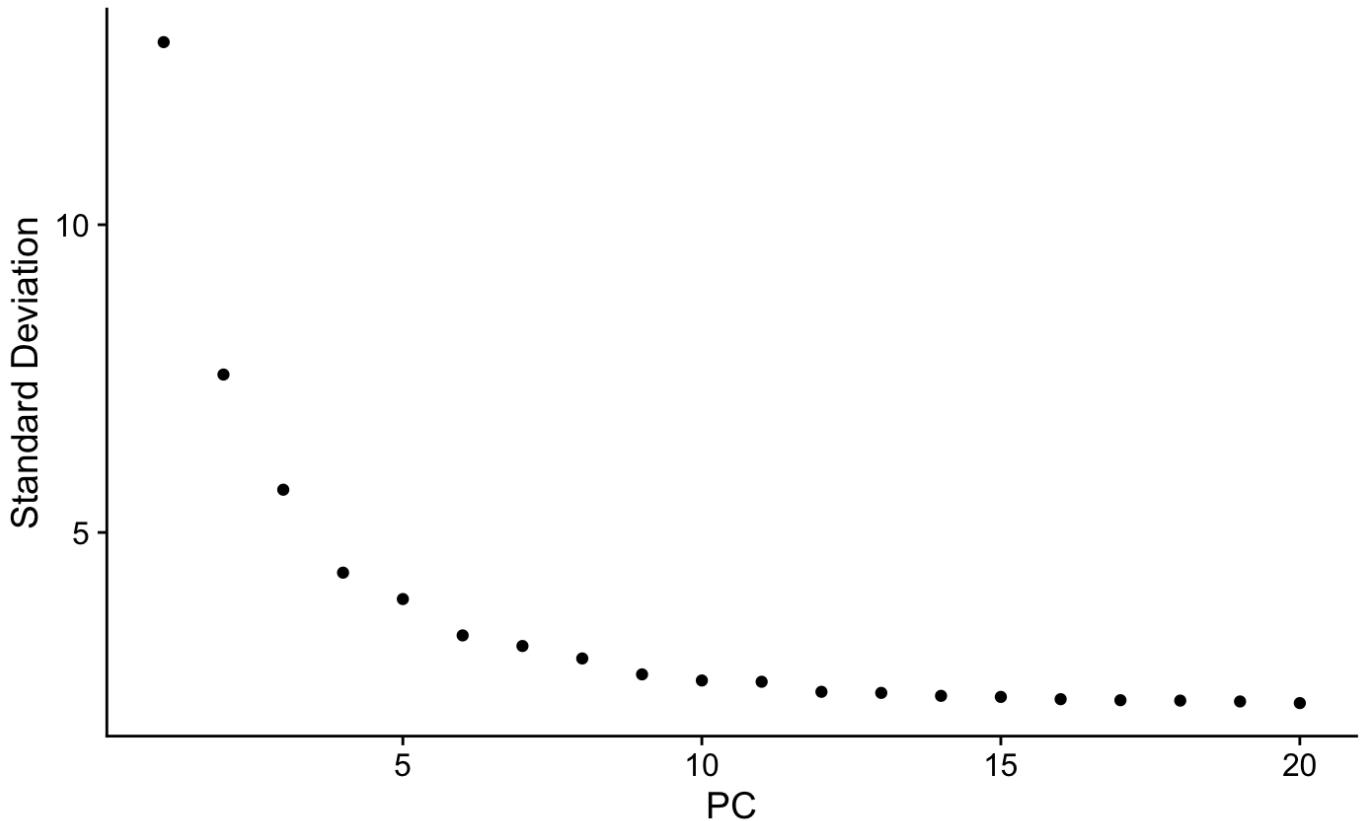
```
#Clean up the data
oligos <- subset(oligos, subset = nFeature_RNA > 200 & nFeature_RNA < 3000 & percent.mt < 10)
```

Optional code to integrate the object (we did not for the paper)

Generating the UMAP and TSNE.

[Hide](#)

```
oligos.integrated <- RunPCA(oligos.integrated, verbose = FALSE)
ElbowPlot(oligos.integrated)
```

[Hide](#)

```
oligos.integrated <- RunUMAP(oligos.integrated, dims = 1:30)
```

```
19:37:54 UMAP embedding parameters a = 0.9922 b = 1.112
19:37:54 Read 3042 rows and found 30 numeric columns
19:37:54 Using Annoy for neighbor search, n_neighbors = 30
19:37:54 Building Annoy index with metric = cosine, n_trees = 50
0%   10    20    30    40    50    60    70    80    90   100%
[----|----|----|----|----|----|----|----|----|
*****|*****|*****|*****|*****|*****|*****|*****|*****|*****|
```

```
19:37:55 Writing NN index file to temp file /var/folders/98/j14x19ln519019fvq32g5rkw0
000gn/T//RtmpCflbt4/file104fd76a18b09
19:37:55 Searching Annoy index using 1 thread, search_k = 3000
19:37:55 Annoy recall = 100%
19:37:56 Commencing smooth kNN distance calibration using 1 thread
19:37:56 Initializing from normalized Laplacian + noise
19:37:59 Commencing optimization for 500 epochs, with 134676 positive edges
0%   10    20    30    40    50    60    70    80    90   100%
[----|----|----|----|----|----|----|----|----|----|
*****|*****|*****|*****|*****|*****|*****|*****|*****|*****|
```

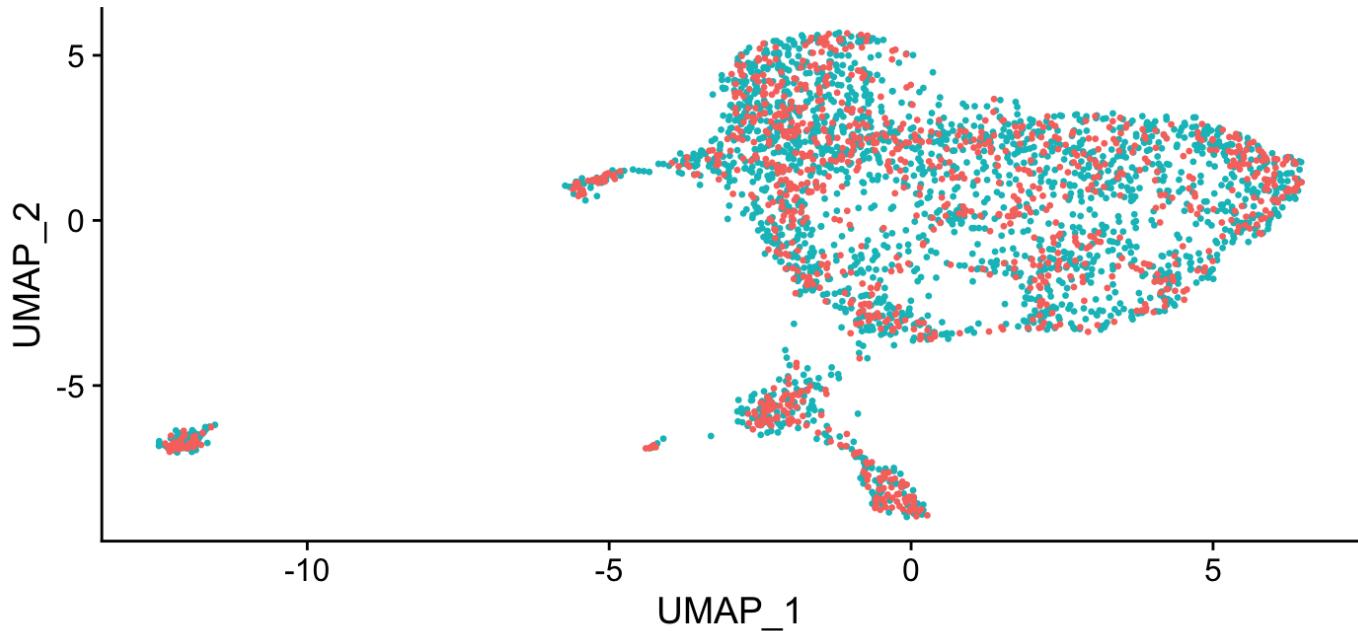
```
19:38:04 Optimization finished
```

[Hide](#)

```
oligos.integrated <- RunTSNE(oligos.integrated, dims = 1:30)
plots <- DimPlot(oligos.integrated, group.by = c("Sample"), combine = FALSE)
plots <- lapply(X = plots, FUN = function(x) x + theme(legend.position = "top") + guides(color = guide_legend(nrow = 3,
  byrow = TRUE, override.aes = list(size = 3))))
CombinePlots(plots)
```

CombinePlots is being deprecated. Plots should now be combined using the patchwork system.

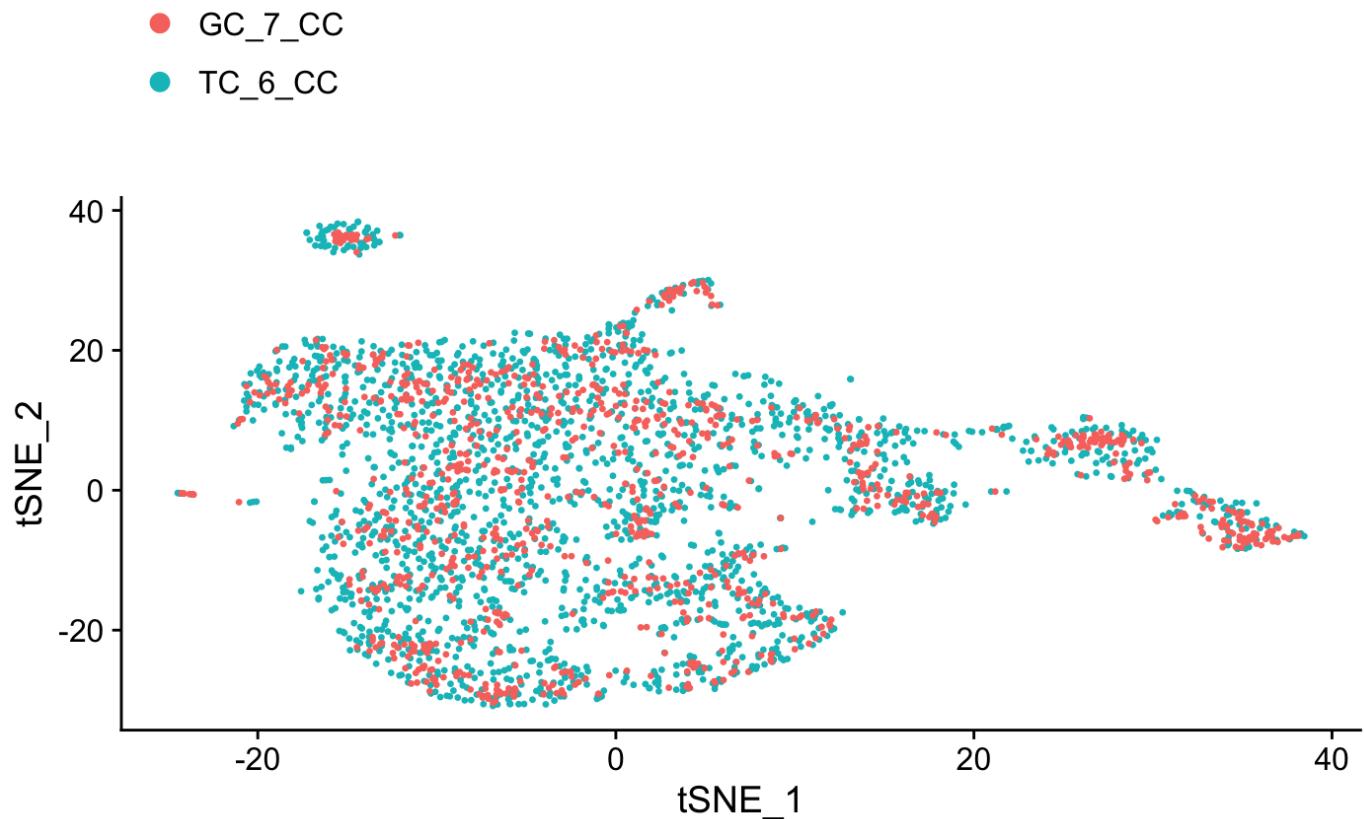
- GC_7_CC
- TC_6_CC



Hide

```
plots <- TSNEPlot(oligos.integrated, group.by = c("Sample"), combine = FALSE)
plots <- lapply(X = plots, FUN = function(x) x + theme(legend.position = "top") + guides(color = guide_legend(nrow = 3,
  byrow = TRUE, override.aes = list(size = 3))))
CombinePlots(plots)
```

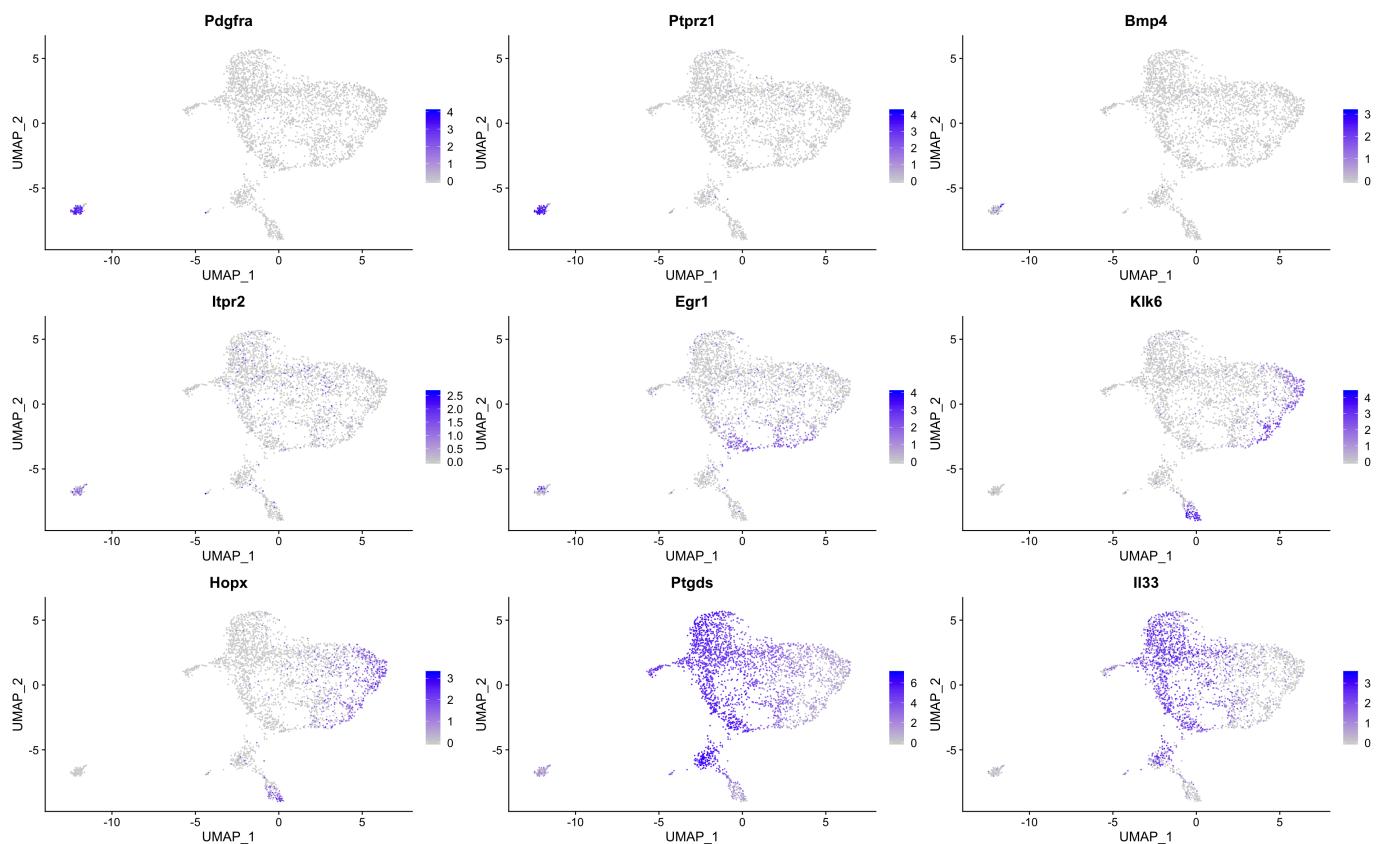
CombinePlots is being deprecated. Plots should now be combined using the patchwork system.



Here I show expression of some common genes that I know are supposed to be more or less stable clusters within the OLs, just for reference.

[Hide](#)

```
DefaultAssay(oligos.integrated) <- "RNA"
# Normalize RNA data for visualization purposes
oligos.integrated <- NormalizeData(oligos.integrated, verbose = FALSE)
FeaturePlot(oligos.integrated, c("Pdgfra", "Ptprz1", "Bmp4", "Itpr2", "Egr1", "Klk6",
"Hopx", "Ptgds", "Il33"), pt.size = 0.1)
```



[Hide](#)

```
DefaultAssay(oligos.integrated) <- "integrated"
```

Here I set the clustering to be specific for at least the tiny cluster of Astrocytes hiding in the middle of the UMAP. COPs are not included, not even with higher clustering resolutions, meaning I can only get them by subclustering. This might be because the COPs are such a tiny cluster in this data, and they express many markers that OPCs are expressing as well.

I show the clusters on the UMAP so you can see their position.

[Hide](#)

```
oligos.integrated <- FindNeighbors(oligos.integrated, dims = 1:30)
```

```
Computing nearest neighbor graph
Computing SNN
```

[Hide](#)

```
oligos.integrated <- FindClusters(oligos.integrated, resolution = 0.8)
```

```
Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
```

```
Number of nodes: 3042
Number of edges: 169348
```

```
Running Louvain algorithm...
```

```
0% 10 20 30 40 50 60 70 80 90 100%
[----|----|----|----|----|----|----|----|----|----|
*****|*****|*****|*****|*****|*****|*****|*****|
```

Maximum modularity in 10 random starts: 0.7834

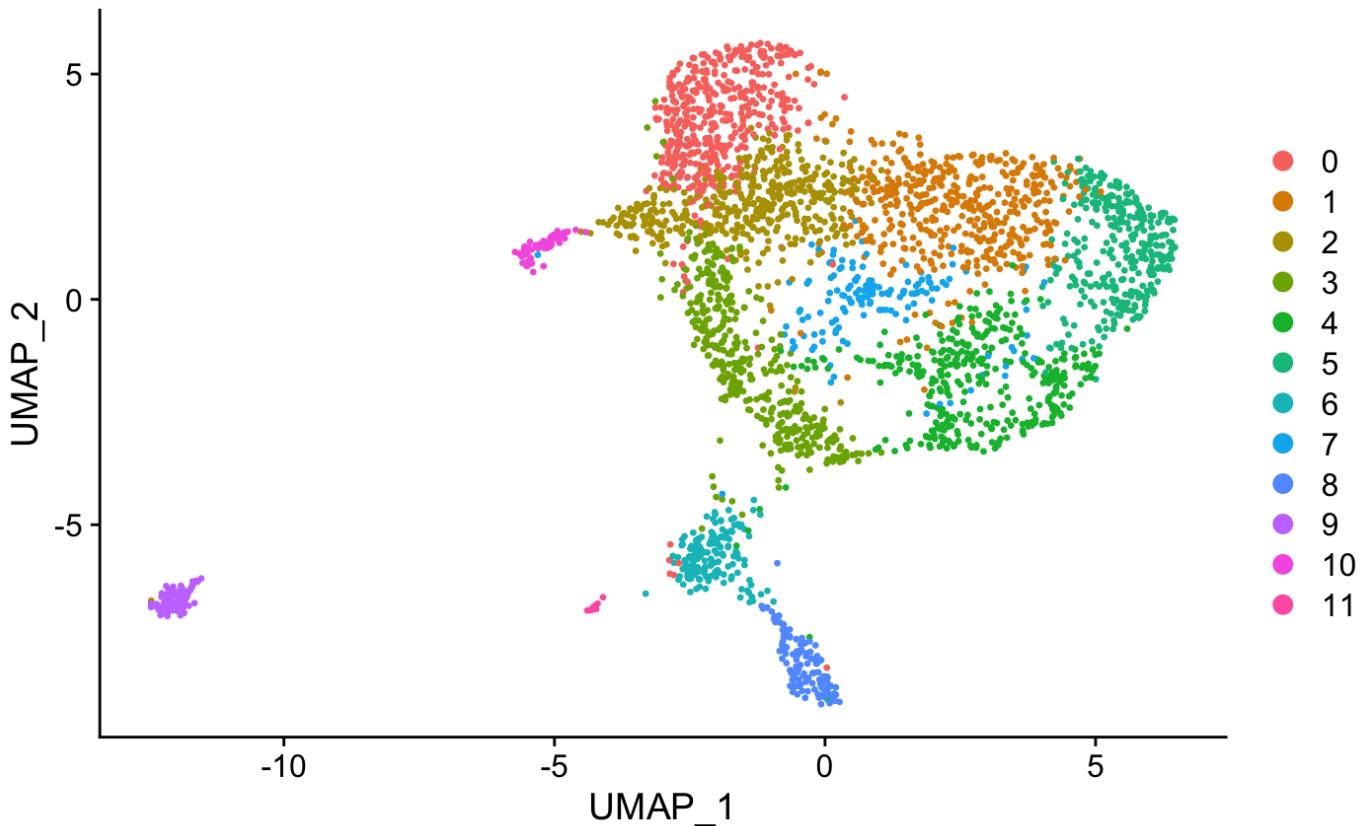
Number of communities: 12

Elapsed time: 0 seconds

[Hide](#)

```
DimPlot(oligos.integrated, group.by = c("seurat_clusters"), combine = FALSE)
```

```
[[1]]
```



Below you will find a table of the top 2 markers found for each cluster. pct means percentage of expression, where pct.2 refers to all the cells not in the tested cluster.

[Hide](#)

```
oligos.integrated.markers %>% group_by(cluster) %>% top_n(n = 2, wt = avg_logFC)
```

p_val	avg_logFC	pct.1	pct.2	p_val_adj	cluster	gene
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<fctr>	<chr>
3.914759e-28	1.807770	0.278	0.101	1.174428e-24	0	Grm7
1.456757e-13	1.859673	0.280	0.157	4.370272e-10	0	Fam13c
8.966872e-31	1.354085	0.625	0.361	2.690062e-27	1	Anln
2.904391e-05	1.612511	0.561	0.416	8.713172e-02	1	Slu7

p_val	avg_logFC	pct.1	pct.2	p_val_adj	cluster	gene
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<fctr>	<chr>
1.181629e-27	1.675580	0.479	0.271	3.544888e-24	2	Sema6d
1.874592e-15	1.747488	0.442	0.297	5.623777e-12	2	Trp53bp2
2.509490e-70	2.399956	0.761	0.362	7.528471e-67	3	Gstp1
1.706142e-23	2.355734	0.557	0.357	5.118426e-20	3	Ubc
3.673240e-80	2.887414	0.781	0.354	1.101972e-76	4	Gapdh
3.762730e-25	2.787885	0.581	0.180	1.128819e-21	4	Ldhb

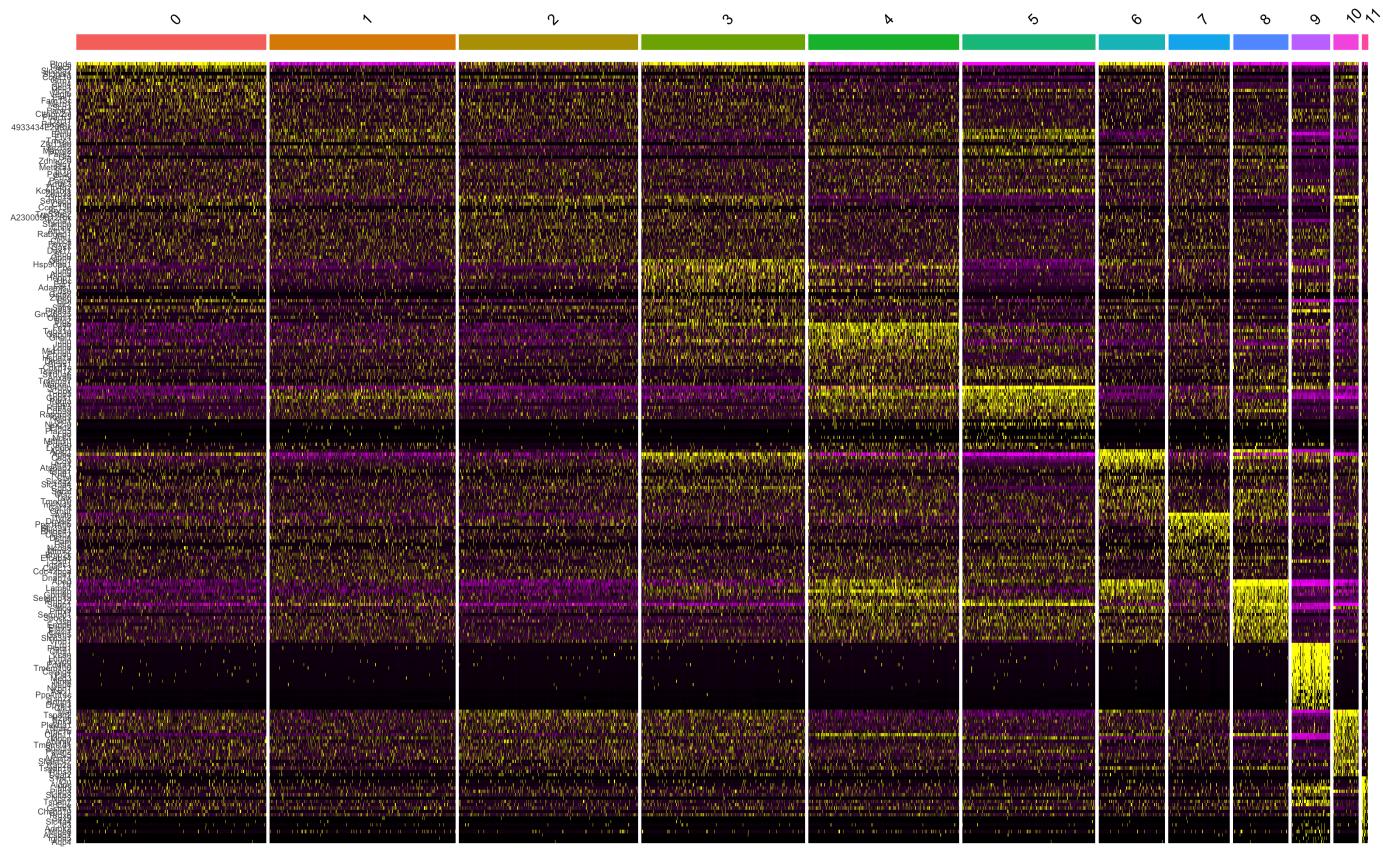
1-10 of 24 rows

Previous **1** 2 3 Next

Below follows the heatmap showing the top 10 genes based on fold change for each cluster.

Hide

```
DefaultAssay(oligos.integrated) <- "SCT"
top10 <- oligos.integrated.markers %>% group_by(cluster) %>% top_n(n = 20, wt = avg_logFC)
DoHeatmap(oligos.integrated, features = top10$gene) + NoLegend()
```



Hide

```
library(viridis)
```

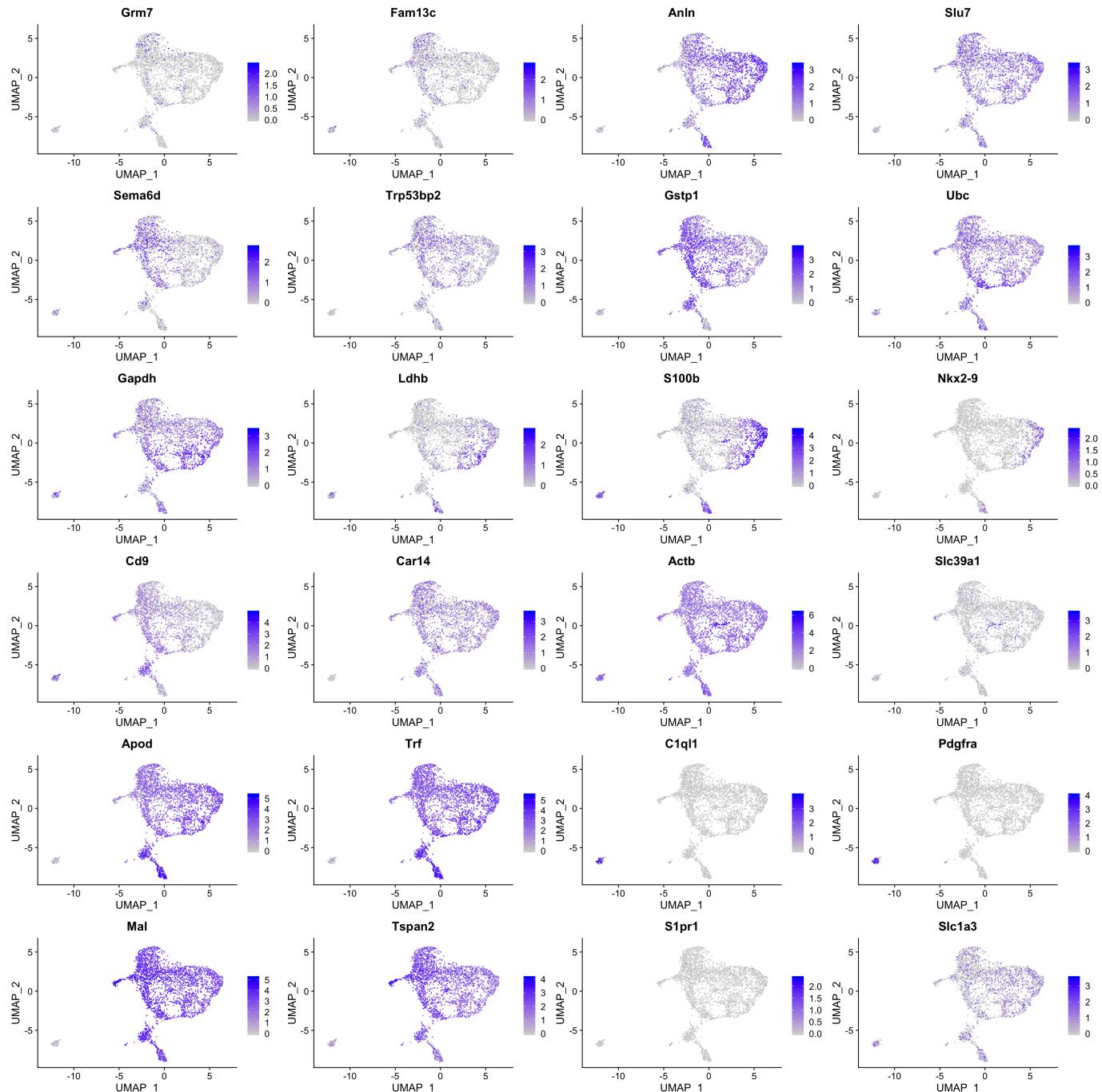
And here are the top 2 genes found for each cluster as show on the UMAP.

Hide

```

DefaultAssay(oligos.integrated) <- "RNA"
# Normalize RNA data for visualization purposes
oligos.integrated <- NormalizeData(oligos.integrated, verbose = FALSE)
top2 <- oligos.integrated.markers %>% group_by(cluster) %>% top_n(n = 2, wt = avg_log
FC)
FeaturePlot(oligos.integrated, features = top2$gene, pt.size = 0.1)

```



Label transfer

Now we attempt to transfer the cluster labels of the Science dataset onto the 10X dataset.

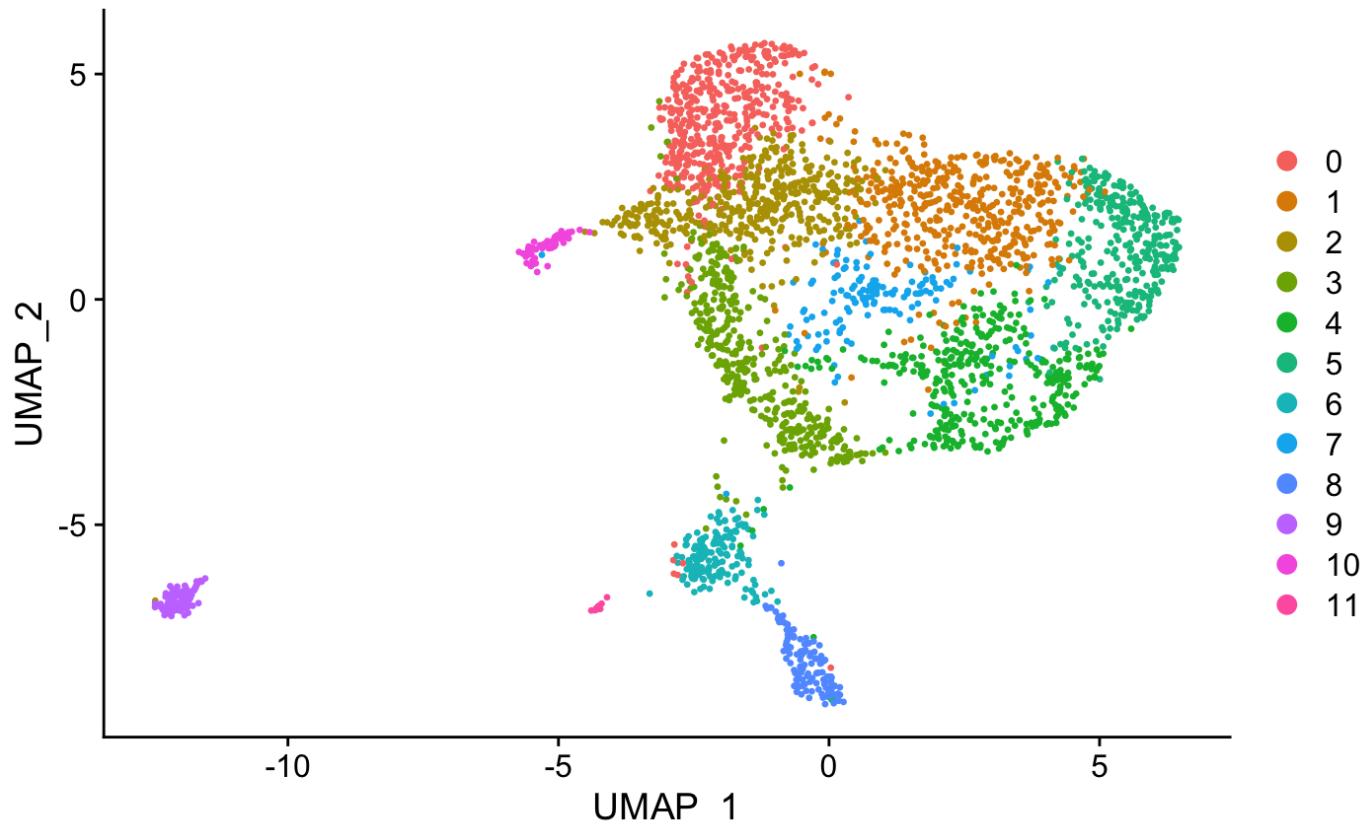
[Hide](#)

```

oligos.integrated$predicted.id <- factor(oligos.integrated$predicted.id, levels=c("OP
C", "COP", "NFOL1", "MFOL1", "MFOL2", "MOL1", "MOL2", "MOL3", "MOL4", "MOL5",
" MOL6"))
DimPlot(oligos.integrated, group.by = c("seurat_clusters"), combine = FALSE)

```

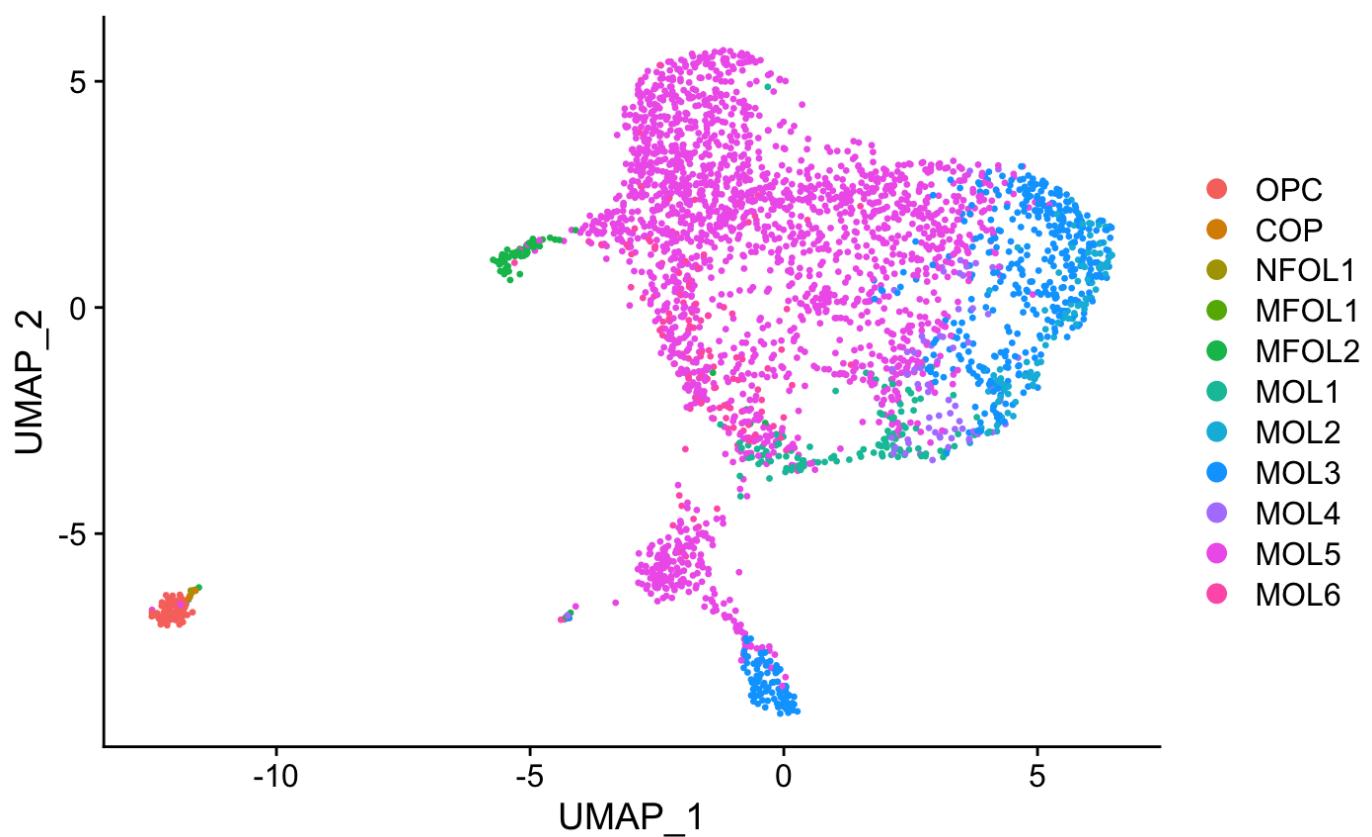
[[1]]



Hide

```
DimPlot(oligos.integrated, group.by = c("predicted.id"), combine = FALSE)
```

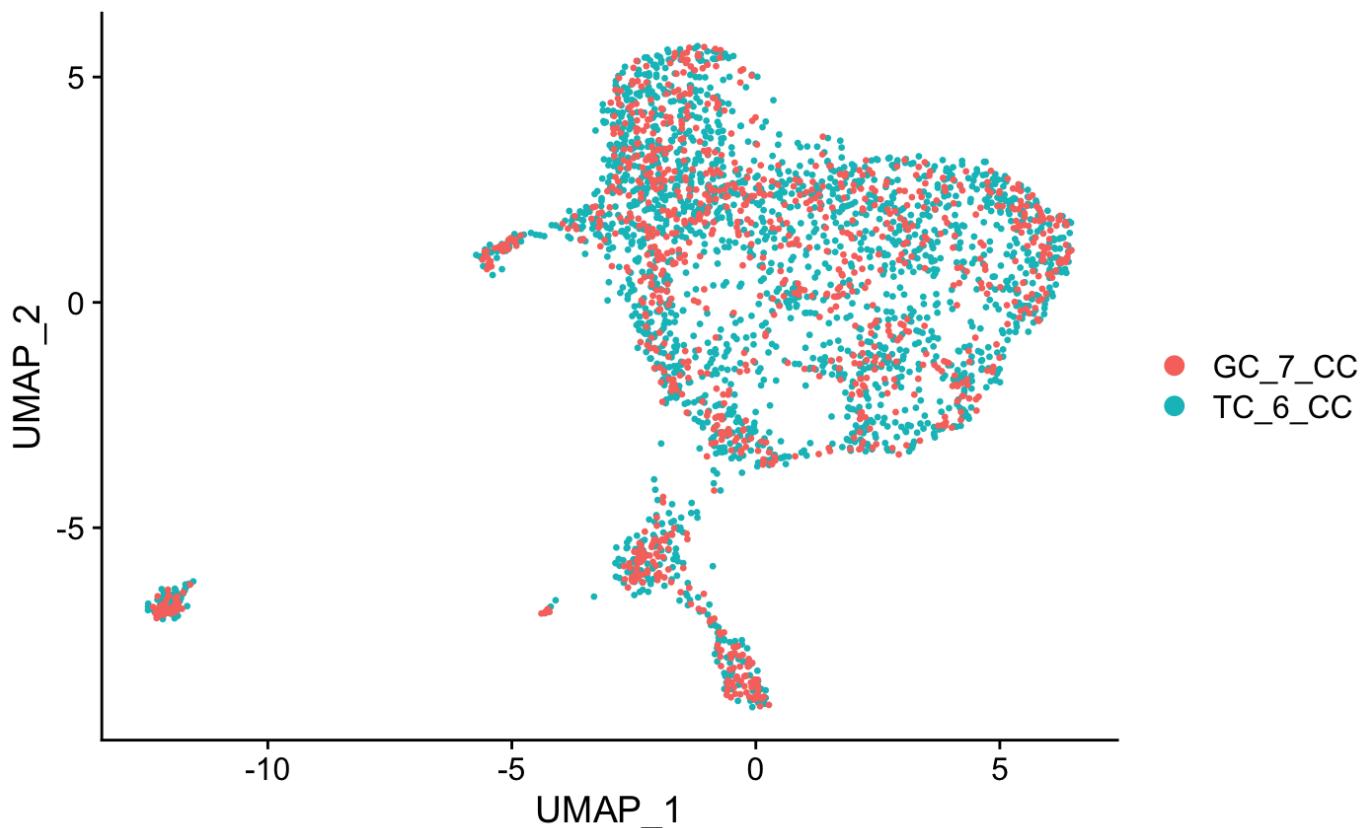
```
[[1]]
```



Hide

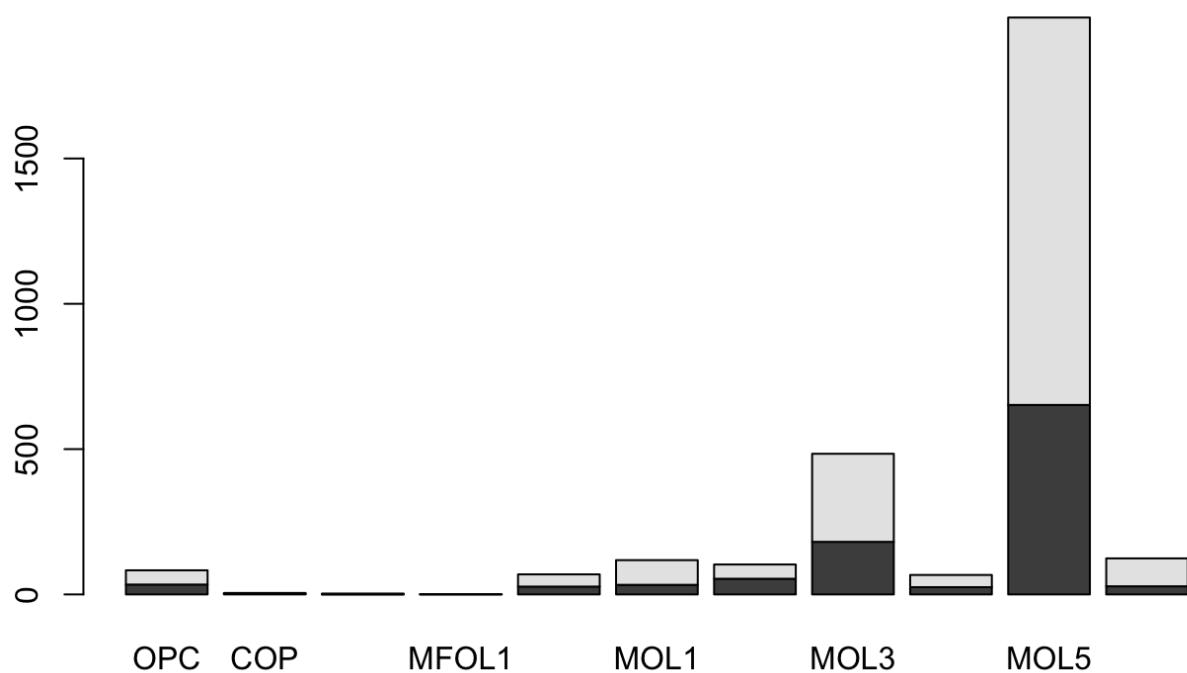
```
DimPlot(oligos.integrated, group.by = c("Sample"), combine = FALSE)
```

```
[[1]]
```



[Hide](#)

```
barplot(table(oligos.integrated$Sample,oligos.integrated$predicted.id))
```



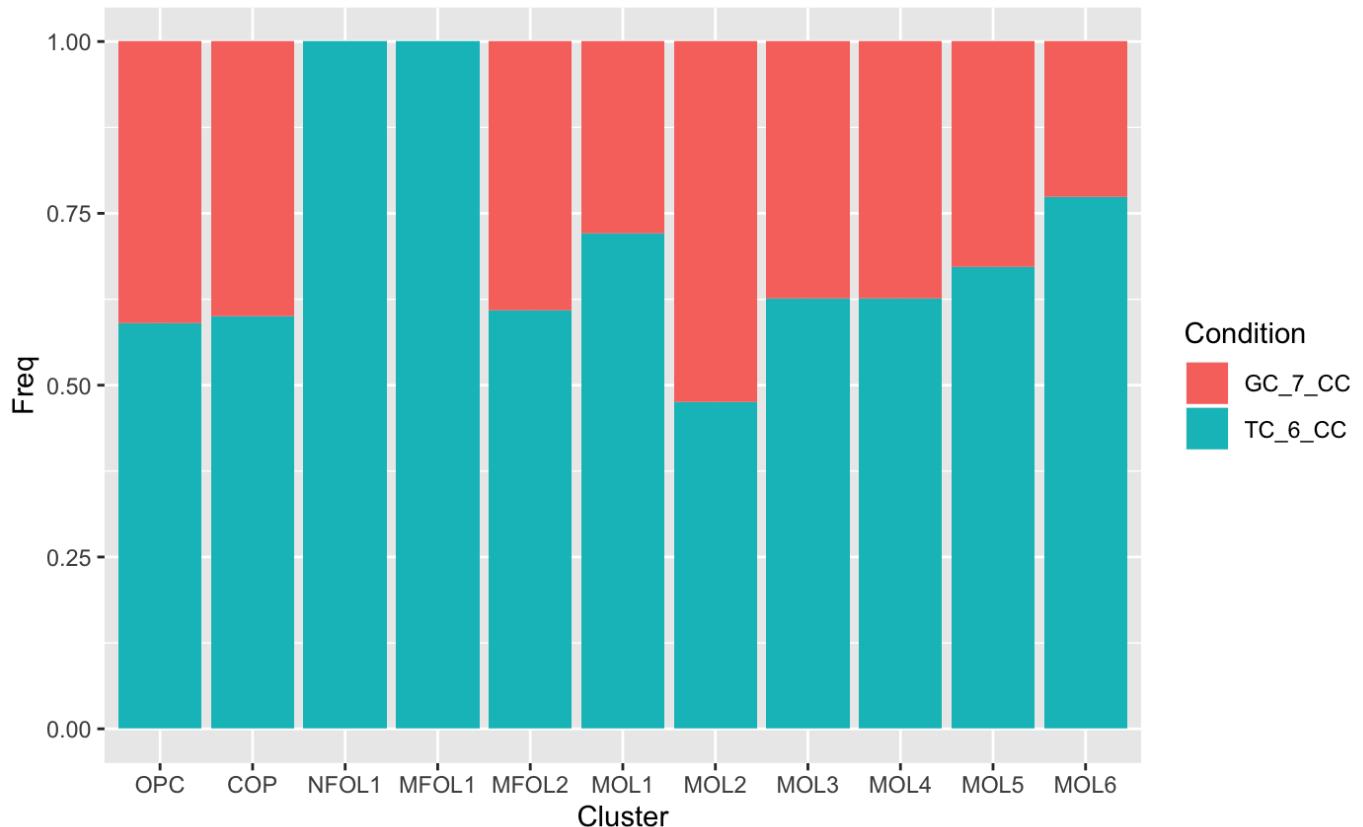
[Hide](#)

```
data <- as.data.frame(table(oligos.integrated$Sample,oligos.integrated$predicted.id))
colnames(data) <- c("Condition","Cluster","Freq")
library(plyr)
data$Cluster  <- factor(data$Cluster,levels=c("OPC","COP","NFOL1","MFOL1","MFOL2","MO
L1","MOL2","MOL3","MOL4","MOL5","MOL6"))
data$Cluster  <- revalue(as.factor(data$Cluster),c("PPR"="VLMC"))
```

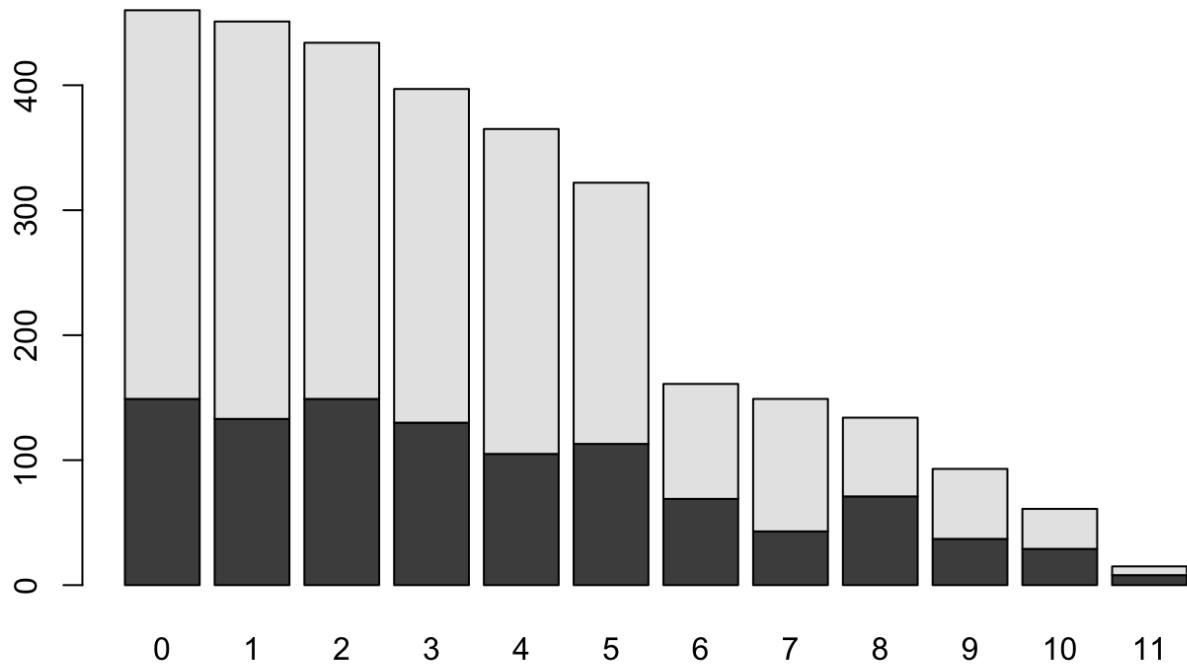
The following `from` values were not present in `x`: PPR

[Hide](#)

```
# Stacked + percent
ggplot(data, aes(fill=Condition, y=Freq, x=Cluster)) +
  geom_bar(position="fill", stat="identity")
```

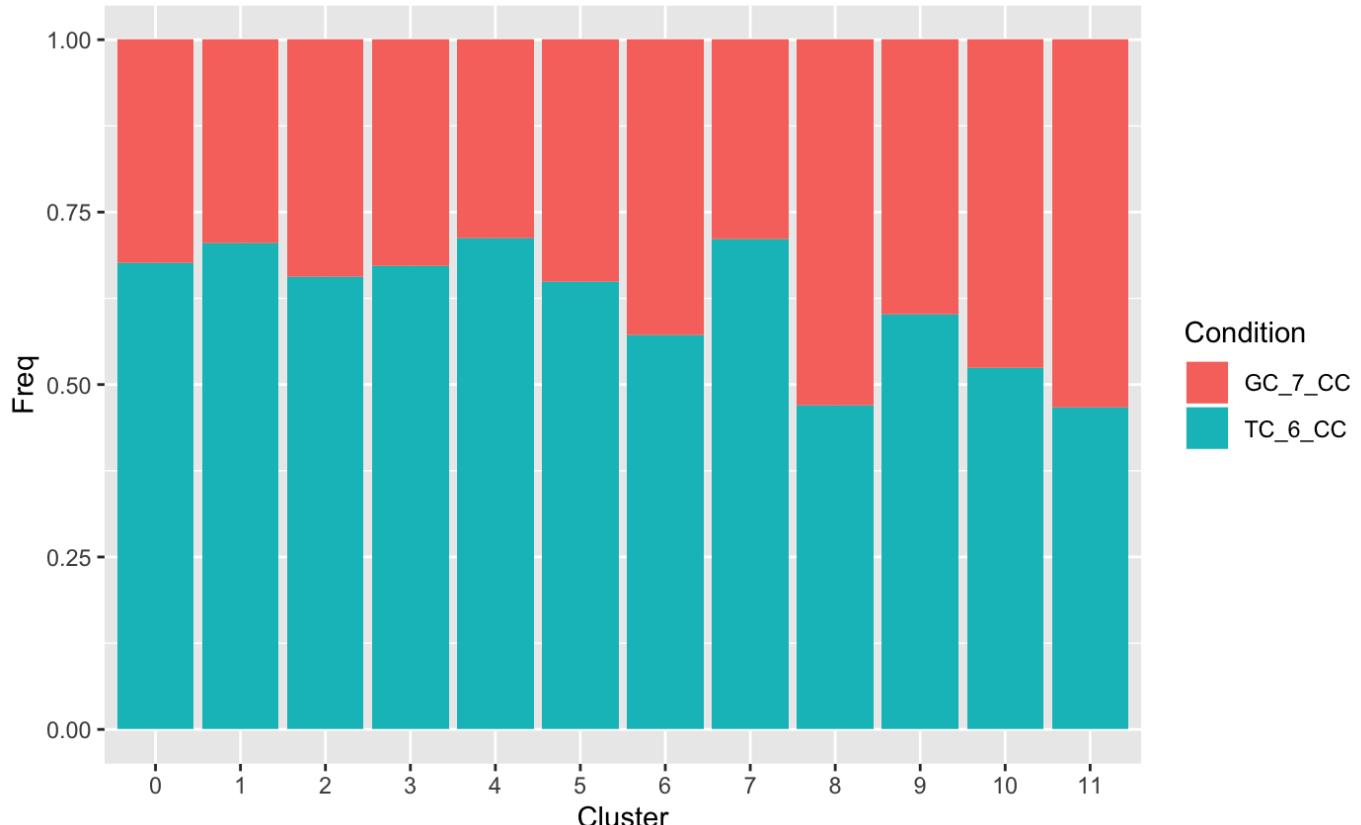
[Hide](#)

```
barplot(table(oligos.integrated$Sample,oligos.integrated$seurat_clusters))
```



[Hide](#)

```
data <- as.data.frame(table(oligos.integrated$Sample,oligos.integrated$seurat_cluster))
colnames(data) <- c("Condition","Cluster","Freq")
library(plyr)
# Stacked + percent
ggplot(data, aes(fill=Condition, y=Freq, x=Cluster)) +
  geom_bar(position="fill", stat="identity")
```



[Hide](#)

```
data <- as.data.frame(table(oligos.integrated$Sample,oligos.integrated$predicted.id))
colnames(data) <- c("Condition","Cluster","Freq")
library(plyr)
data$Cluster  <- factor(data$Cluster,levels=c("OPC","COP","NFOL1","MFOL1","MFOL2","MO
L1","MOL2","MOL3","MOL4","MOL5","MOL6"))
library(reshape2)
datacasted <- dcast(data,Cluster ~ Condition)
```

Using Freq as value column: use value.var to override.

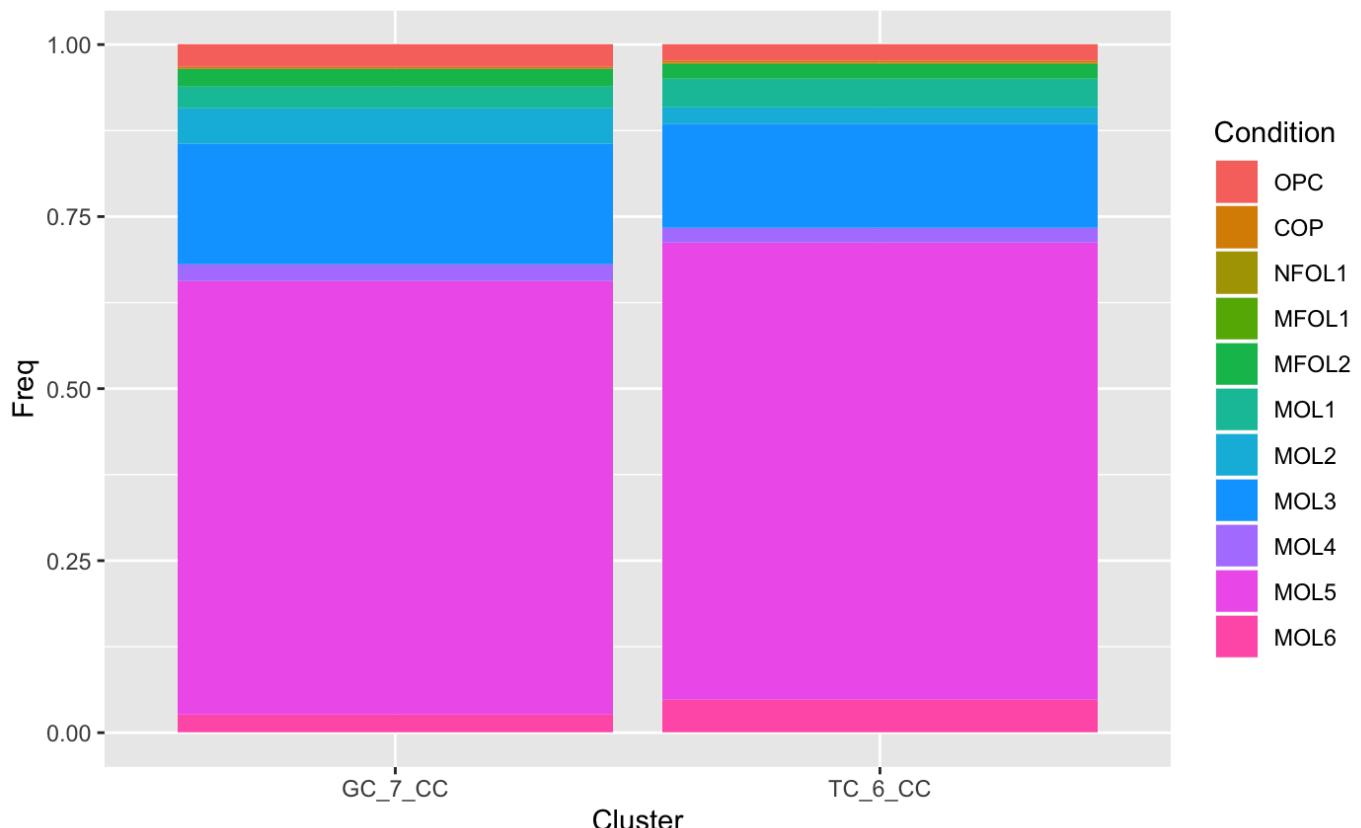
[Hide](#)

```
calc_cpm <-function (expr_mat)
{
  norm_factor <- colSums(expr_mat)
  return(t(t(expr_mat)/norm_factor)) * 10^50
}
datacasted[,2:3] <- calc_cpm(datacasted[,2:3])
data <- melt(datacasted)
```

Using Cluster as id variables

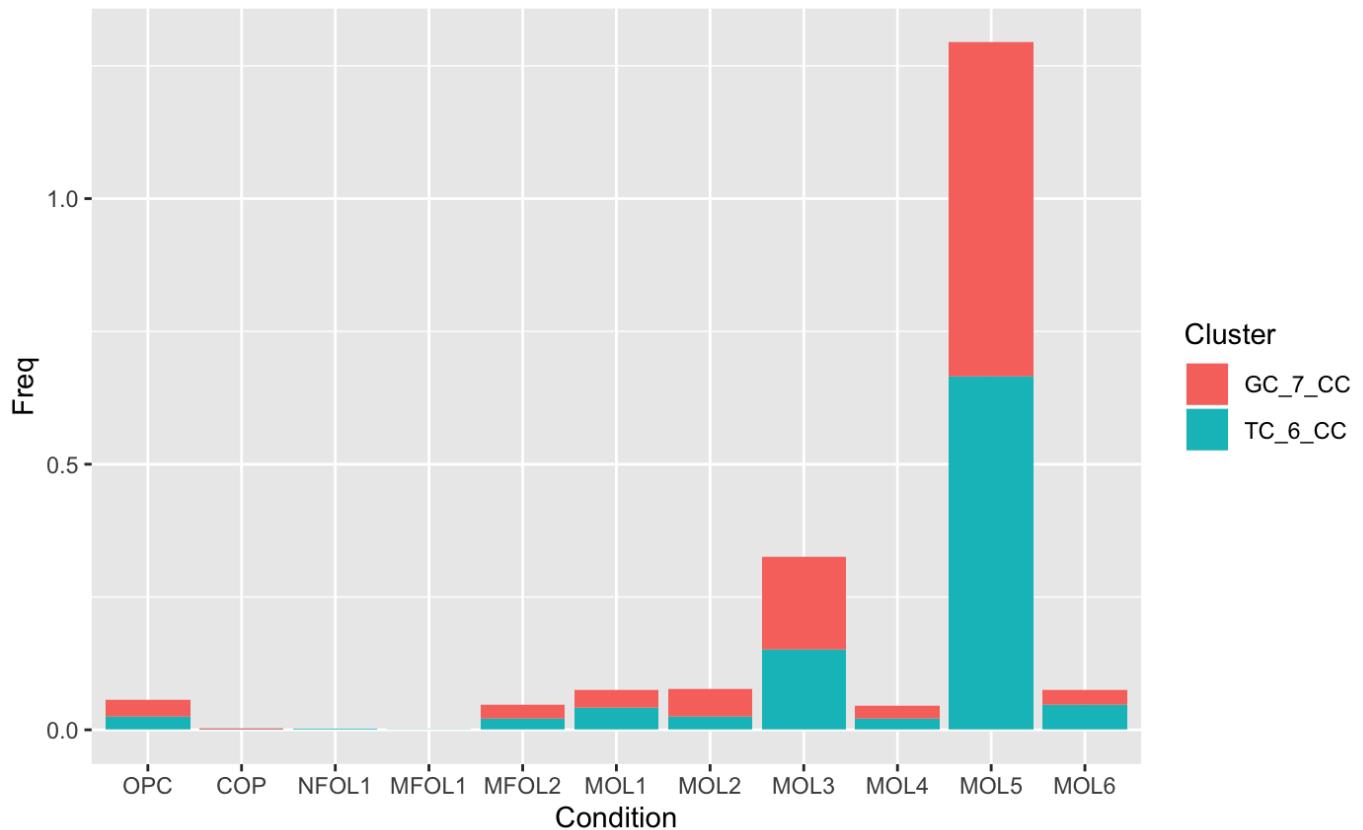
[Hide](#)

```
colnames(data) <- c("Condition","Cluster","Freq")
#data$Cluster <- revalue(as.factor(data$Cluster),c("PPR"="VLMC"))
# Stacked + percent
ggplot(data, aes(fill=Condition, y=Freq, x=Cluster)) +
  geom_bar(position="fill", stat="identity")
```

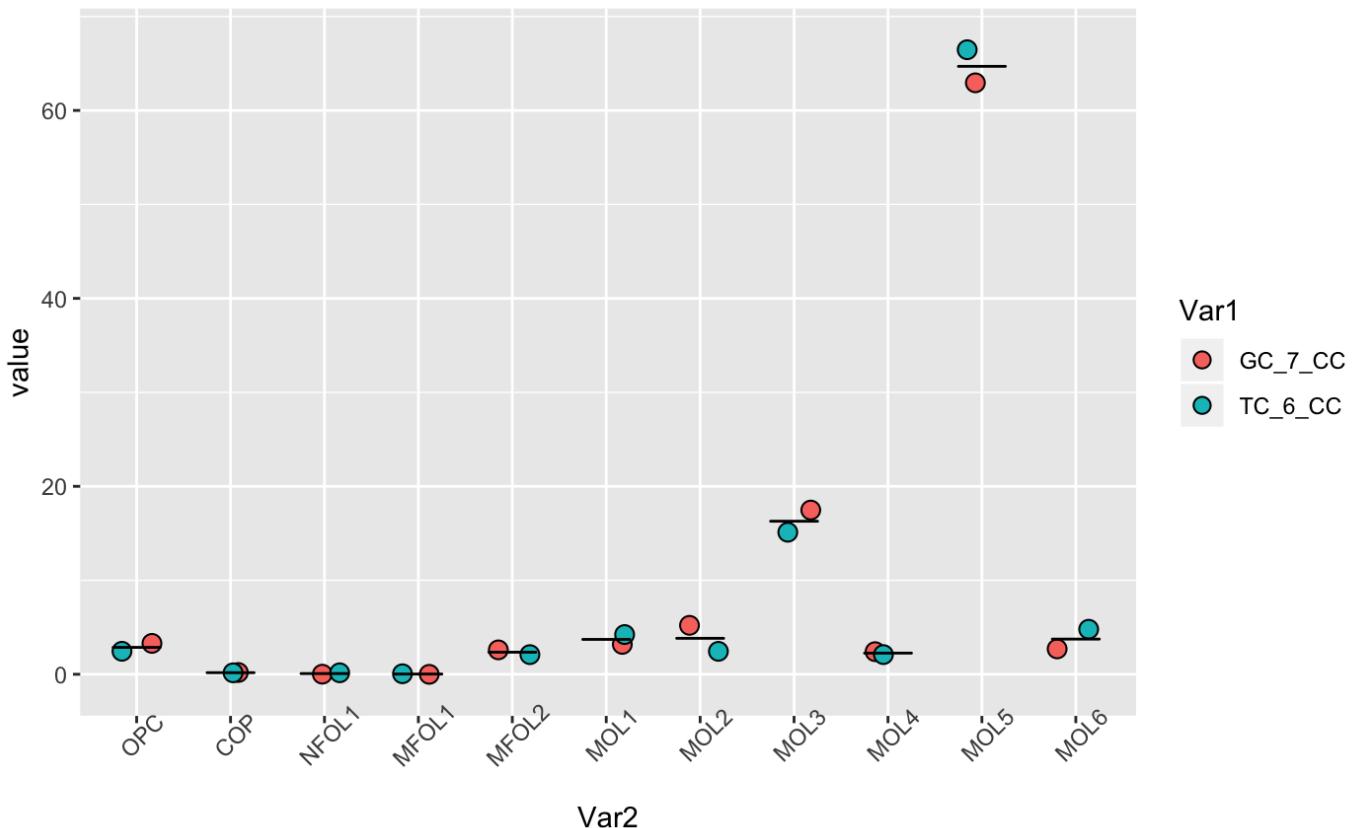


[Hide](#)

```
ggplot(data, aes(fill=Cluster, y=Freq, x=Condition)) +  
  geom_bar( stat="identity")
```

[Hide](#)

```
row.names(datacasted) <- datacasted[,1]  
datacasted <- datacasted[,2:3]*100  
datamelted <- melt(t(datacasted))  
ggplot(datamelted, aes(y = value, x = Var2)) + # Move y and x here so than they can b  
e used in stat_*  
  geom_dotplot(aes(fill = Var1),    # Use fill = Species here not in ggplot()  
               binaxis = "y",          # which axis to bin along  
               binwidth = 2,           # Minimal difference considered different  
               stackdir = "center",  
               position = position_jitter(0.2)# Centered  
               ) + # scale_y_log10() +  
  stat_summary(fun.y = mean, fun.ymin = mean, fun.ymax = mean,  
              geom = "crossbar", width = 0.5,fatten = 0.01) + theme(axis.text.x =  
element_text(angle = 45))
```



```
oligos.integrated.markers %>% group_by(cluster) %>% top_n(n = 2, wt = avg_logFC)
```

p_val	avg_logFC	pct.1	pct.2	p_val_adj	cluster	gene
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<fctr>	<chr>
0.000000e+00	2.3298483	0.988	0.020	0.000000e+00	OPC	Ptprz1
6.052966e-249	2.4776942	0.988	0.049	7.408225e-245	OPC	Cspg5
5.415098e-36	2.8766542	1.000	0.031	6.627538e-32	COP	Gpr17
6.372811e-10	2.4303309	1.000	0.152	7.799683e-06	COP	Lims2
9.025865e-21	2.5829249	1.000	0.035	1.104676e-16	NFOL1	Rtn1
1.639679e-18	2.3098033	1.000	0.038	2.006804e-14	NFOL1	Marcks
3.972733e-07	2.3098033	1.000	0.133	4.862229e-03	NFOL1	Mpzl1
1.164613e-31	1.1197770	0.971	0.882	1.425369e-27	MFOL2	Tspan2
4.492967e-30	1.0369061	0.986	0.979	5.498942e-26	MFOL2	Mal
6.530282e-101	1.7776452	0.958	0.225	7.992412e-97	MOL1	Fos

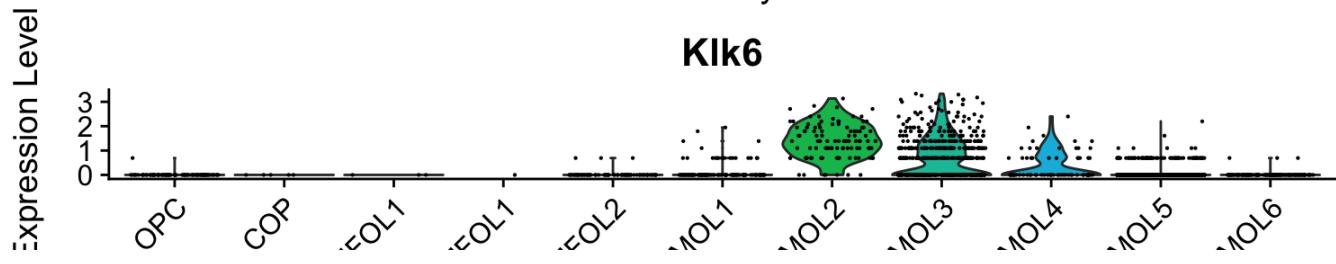
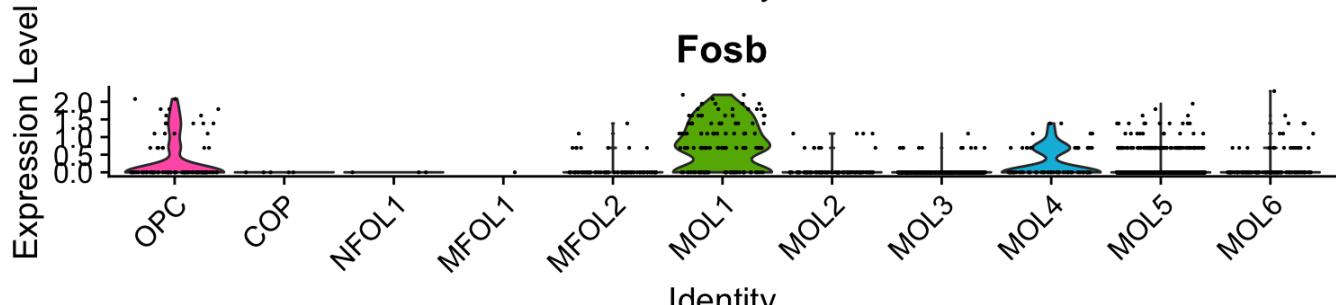
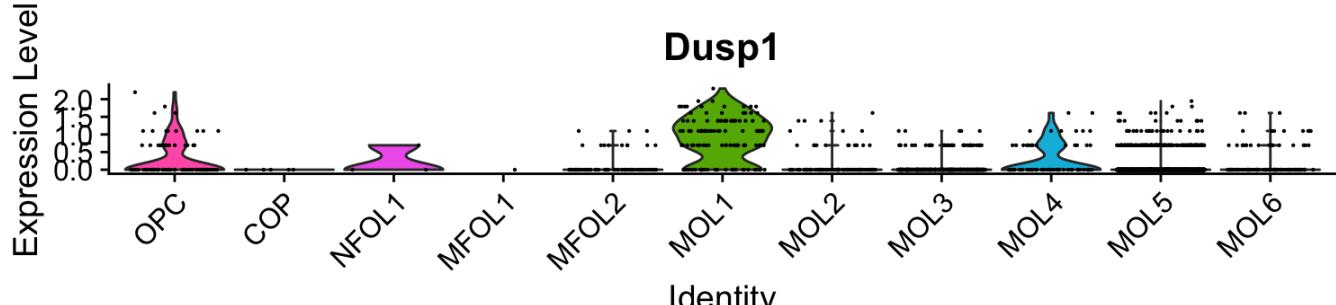
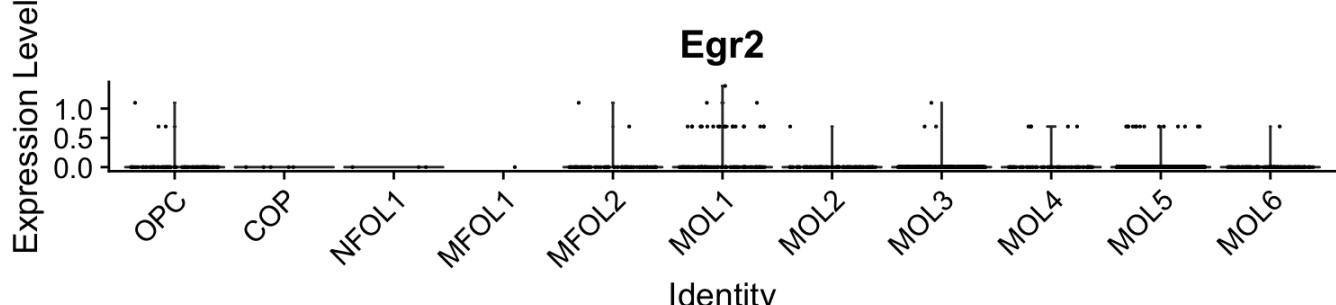
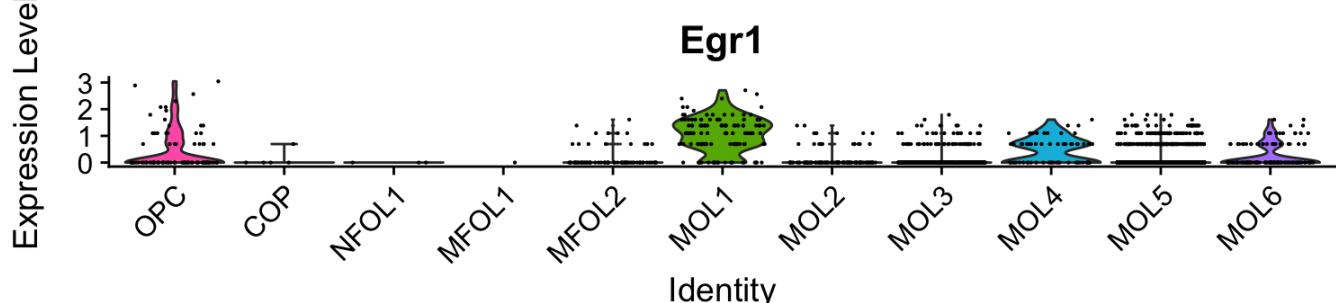
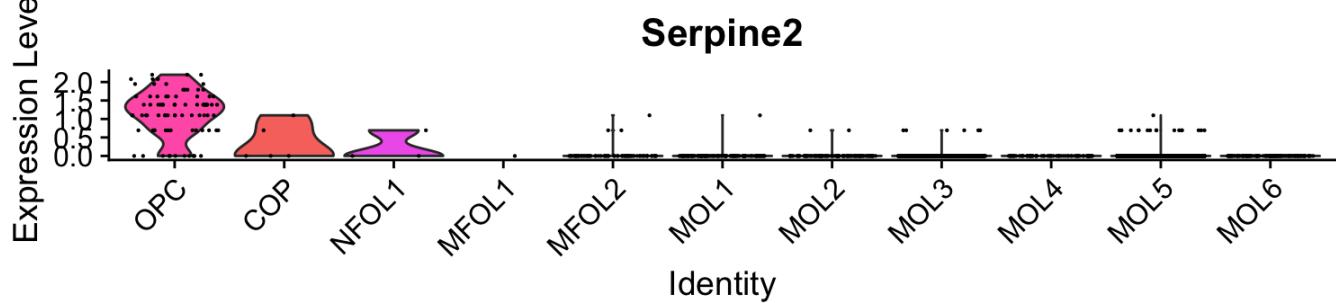
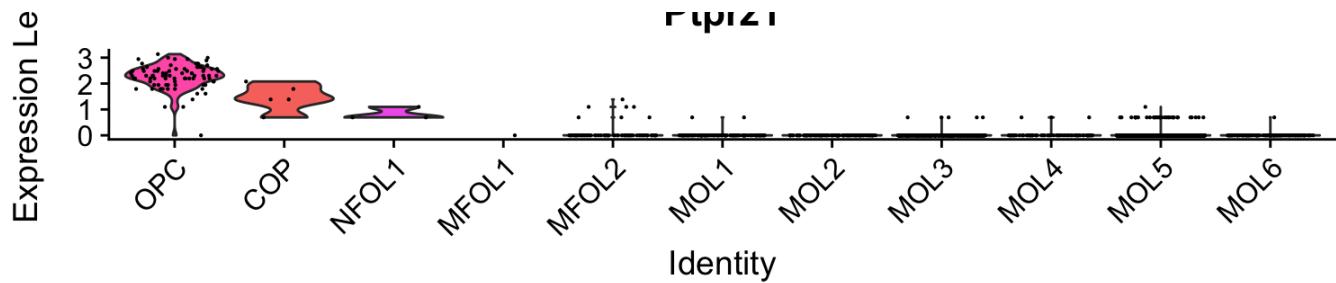
1-10 of 21 rows

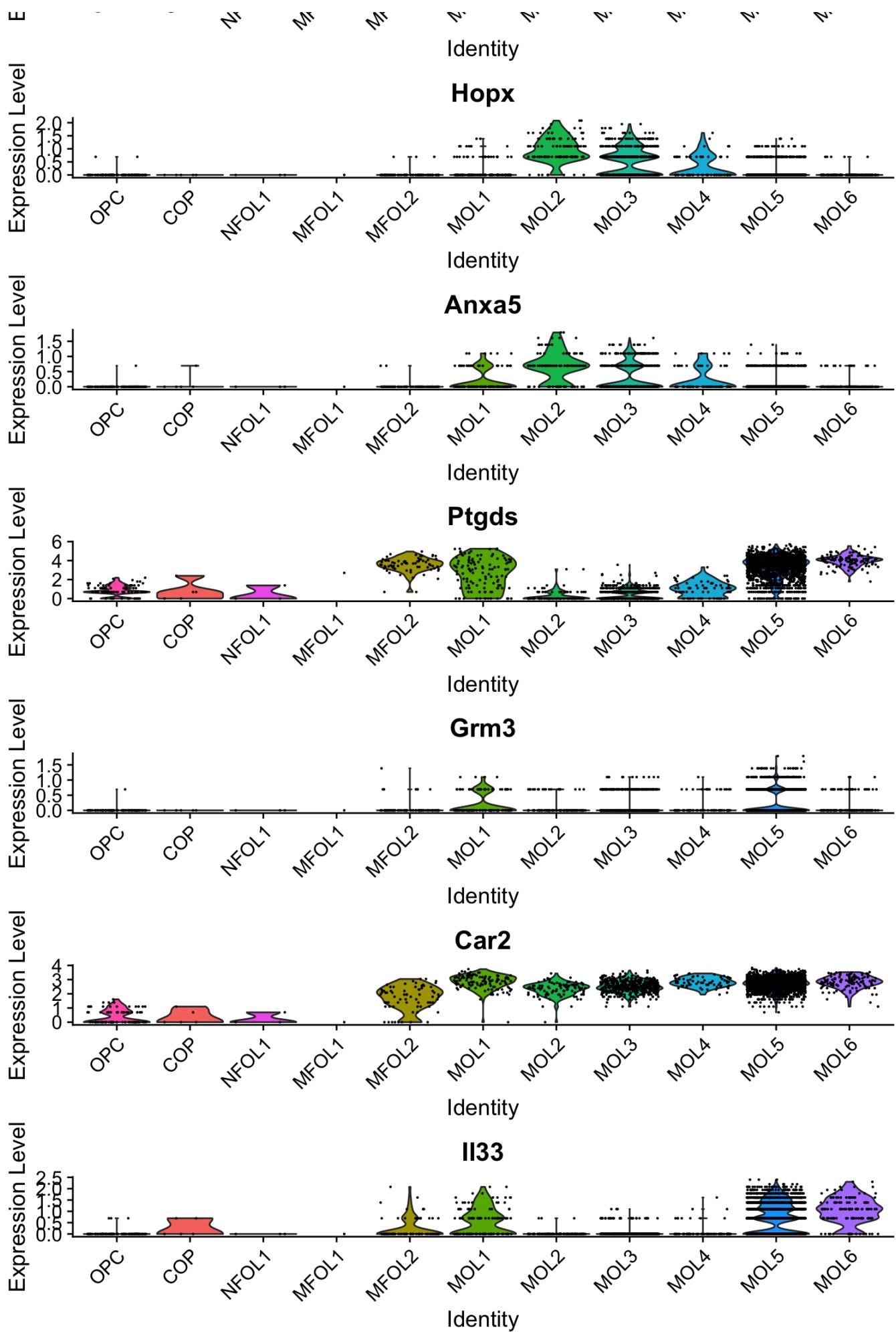
Previous **1** 2 3 Next

```
VlnPlot(oligos.integrated, group.by = "predicted.id", features = c("Ptprz1", "Serpine2", "Egr1", "Egr2", "Dusp1", "Fosb", "Klk6", "Hopx", "Anxa5", "Ptgds", "Grm3", "Car2", "Il33"), ncol = 1, pt.size = 0.1)
```

vel

Dtors1





Hide

```

library(viridis)
DefaultAssay(oligos.integrated) <- "integrated"
top10 <- oligos.integrated.markers %>% group_by(cluster) %>% top_n(n = 10, wt = avg_1
ogFC)
DoHeatmap(oligos.integrated, features = top10$gene) + NoLegend() + scale_fill_viridis()

```

Scale for 'fill' is already present. Adding another scale for 'fill', which will replace the existing scale.

