

Preliminary spacial transcriptomic analysis

Jieun Jeong

2025-03-18

Initial steps

Loading libraries and data.

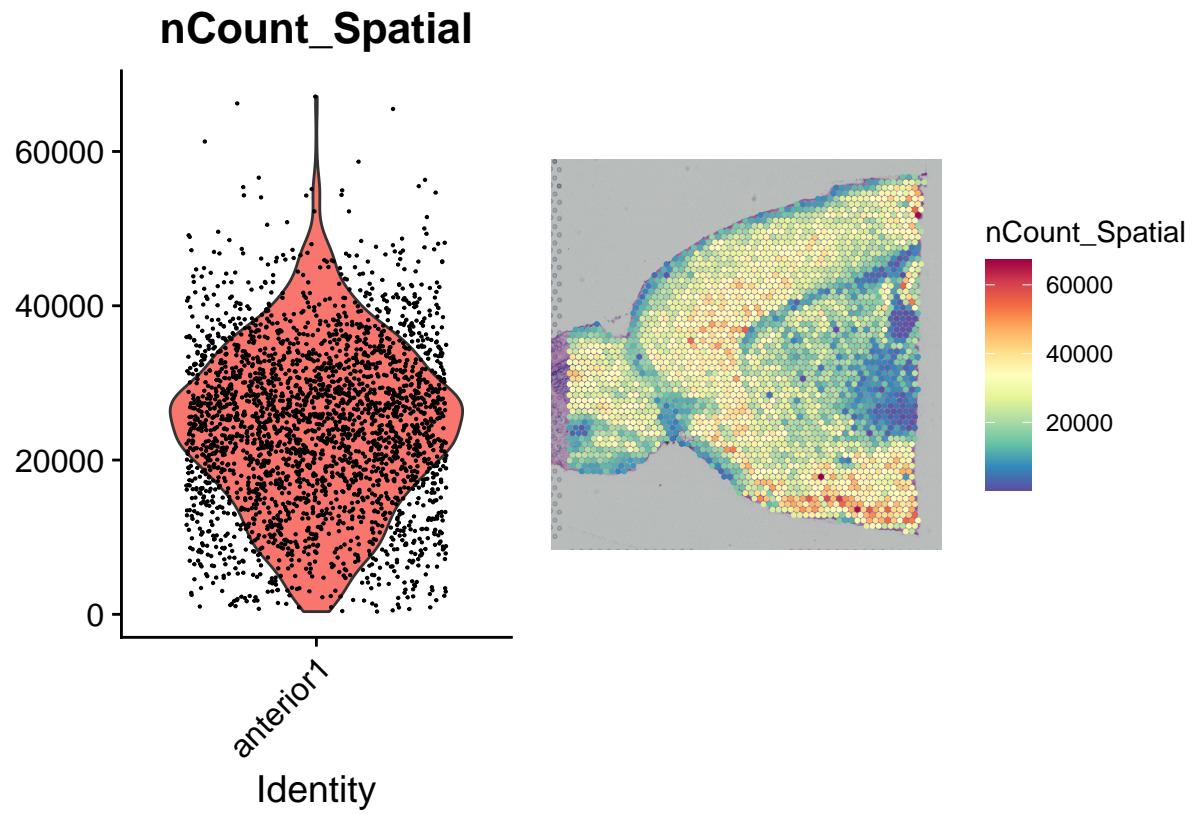
```
pa = "" # change it if you have a path to a directory described above
library(dplyr)
library(Seurat)
library(patchwork)
library(ggplot2)
```

This is how we get example data:

```
if (!requireNamespace("remotes", quietly = TRUE)) {
  install.packages("remotes")
}
if (!requireNamespace("SeuratData", quietly = TRUE)) {
  remotes::install_github("satijalab/seurat-data")
}
library(SeuratData)
InstallData("stxBrain")
brain <- LoadData("stxBrain", type = "anterior1")
```

It is always good to visualize basic properties of the data, and it is especially important for spacial data

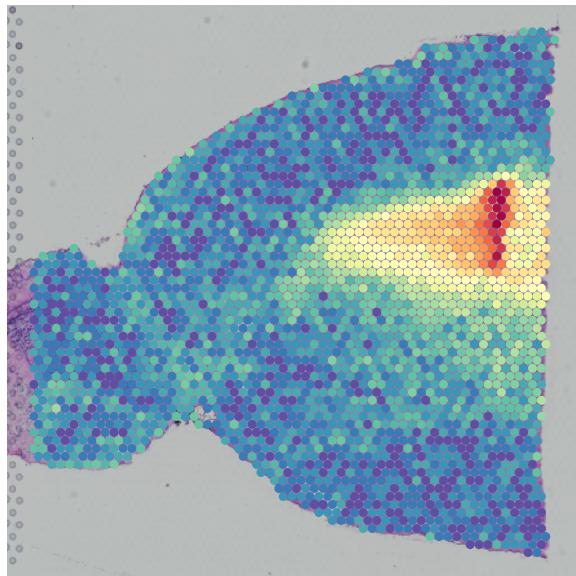
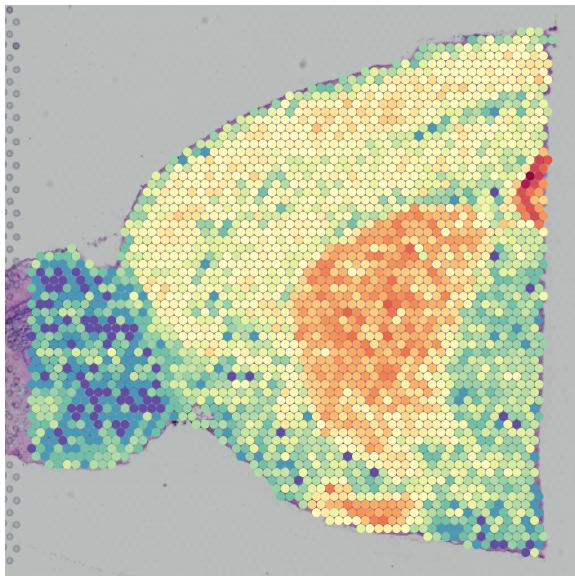
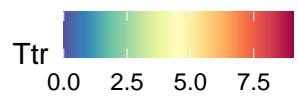
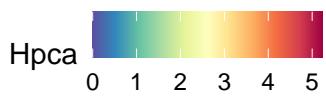
```
plot1 <- VlnPlot(brain, features = "nCount_Spatial", pt.size = 0.1) + NoLegend()
plot2 <- SpatialFeaturePlot(brain, features = "nCount_Spatial", pt.size.factor = 2.6) + theme(legend.position = "none")
wrap_plots(plot1, plot2)
```



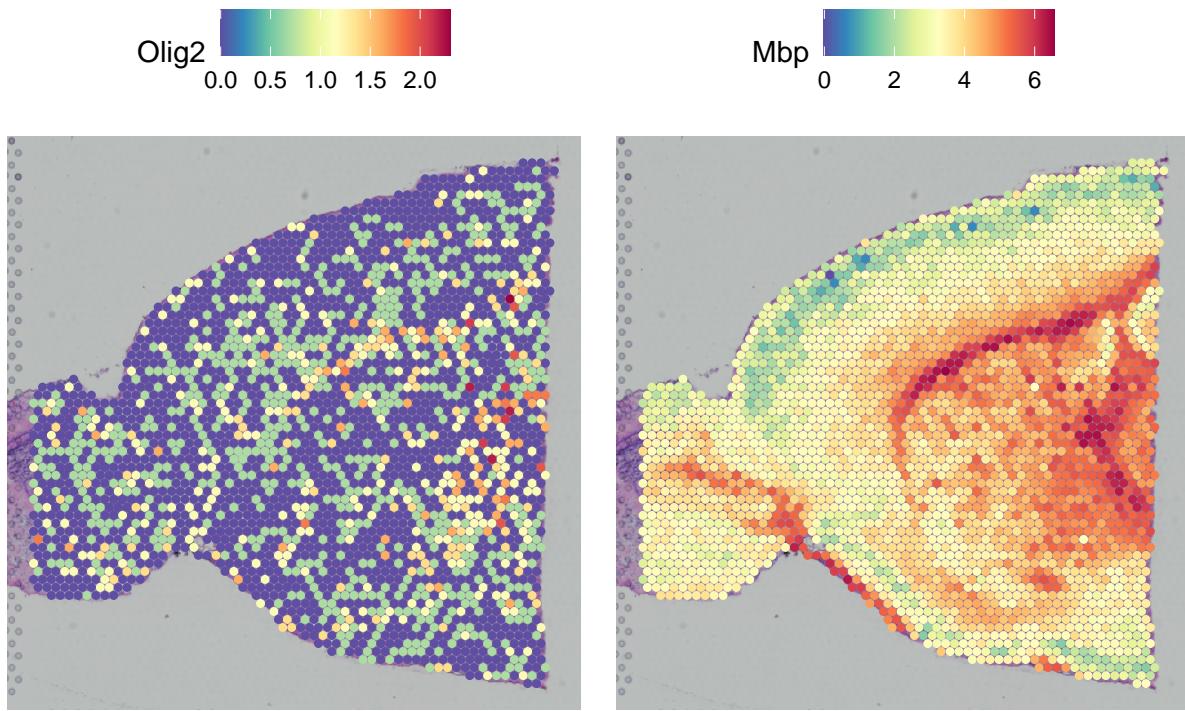
The counts per hexagonal pixels drop at the boundary, and is lower in white matter where we have axons and nuclei of astrocytes, blue in center right, and higher in gray matter that surrounds it, yellow to red hues.

Comparing localization of markers. Below left, hippocampus marker Hpc and on the right, choroid plexus marker Ttr.

```
brain <- SCTtransform(brain, assay = "Spatial", verbose = FALSE)
SpatialFeaturePlot(brain, features = c("Hpc", "Ttr"), pt.size.factor = 2.7)
```



```
SpatialFeaturePlot(brain, features = c("Olig2", "Mbp"), pt.size.factor = 2.7)
```

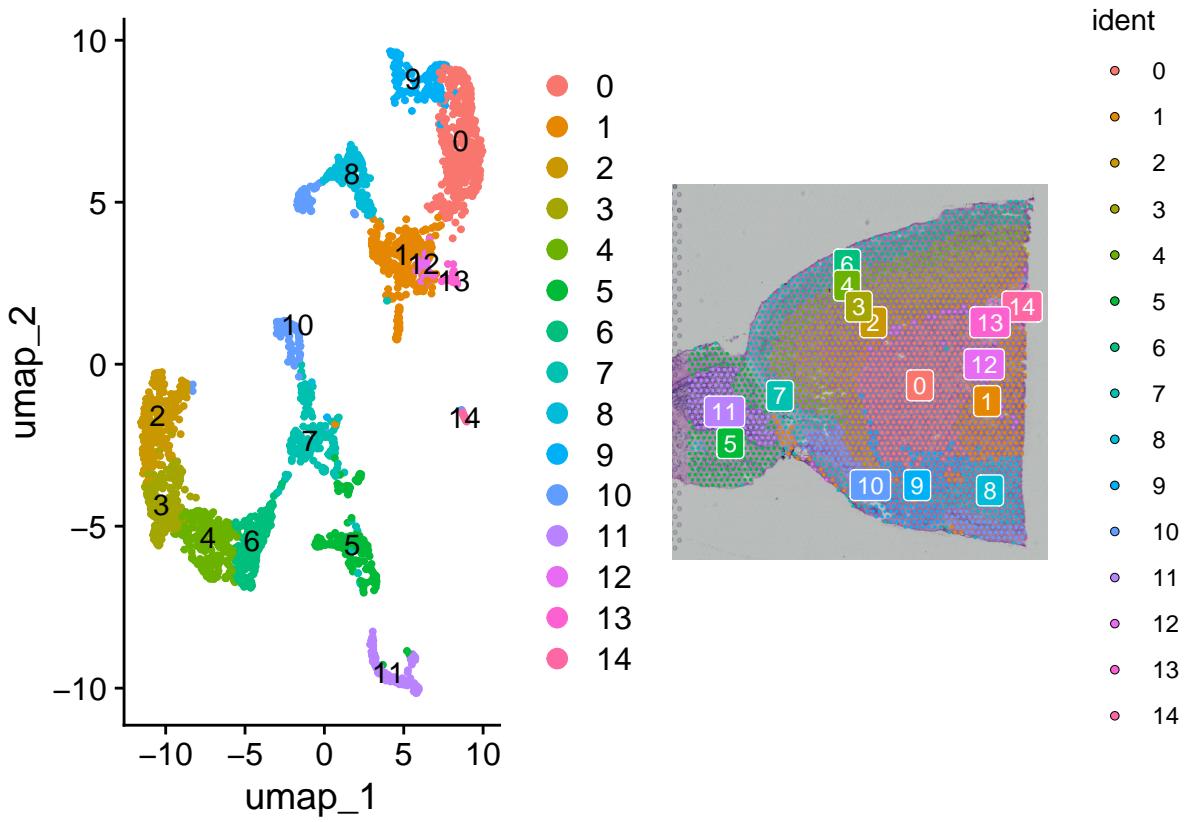


PCA, finding clusters of voxels We can identify different cell populations, which may be heterogenous, and their contacts. PCA reduces the dimension from many thousands to 30, we cluster in this reduced space and then we find UMAP coordinates to visualize the clusters.

```
brain <- RunPCA(brain, assay = "SCT", verbose = FALSE)
brain <- FindNeighbors(brain, reduction = "pca", dims = 1:30)
brain <- FindClusters(brain, verbose = FALSE)
brain <- RunUMAP(brain, reduction = "pca", dims = 1:30)
```

Now the preliminary visualization:

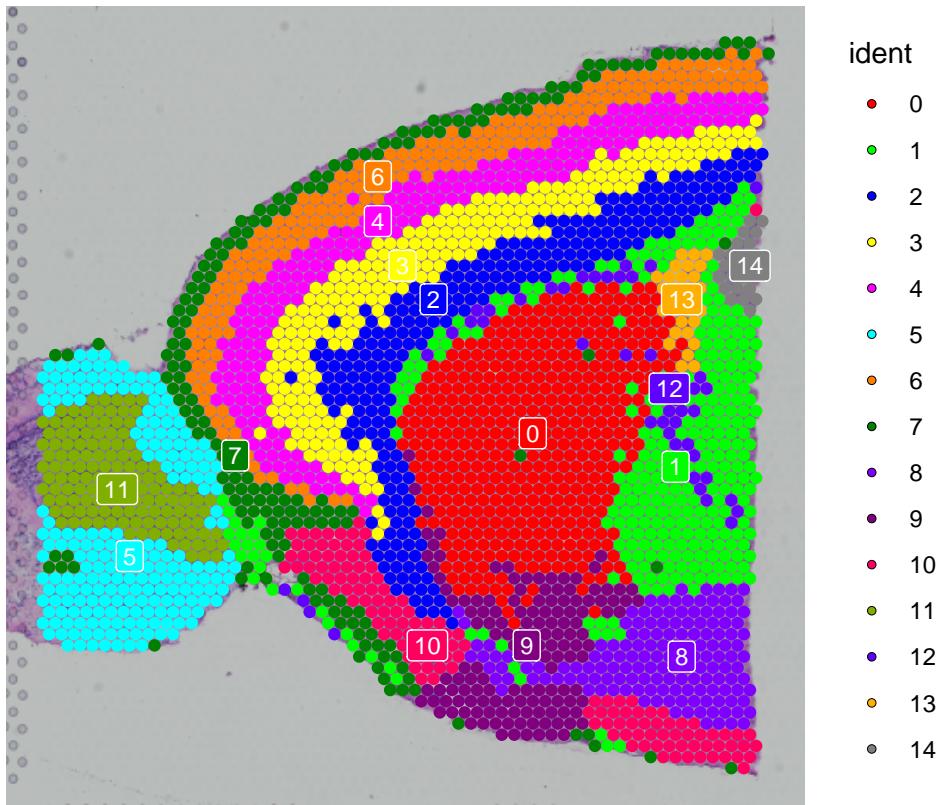
```
p1 <- DimPlot(brain, reduction = "umap", label = TRUE)
p2 <- SpatialDimPlot(brain, label = TRUE, label.size = 3)
p1 + p2
```



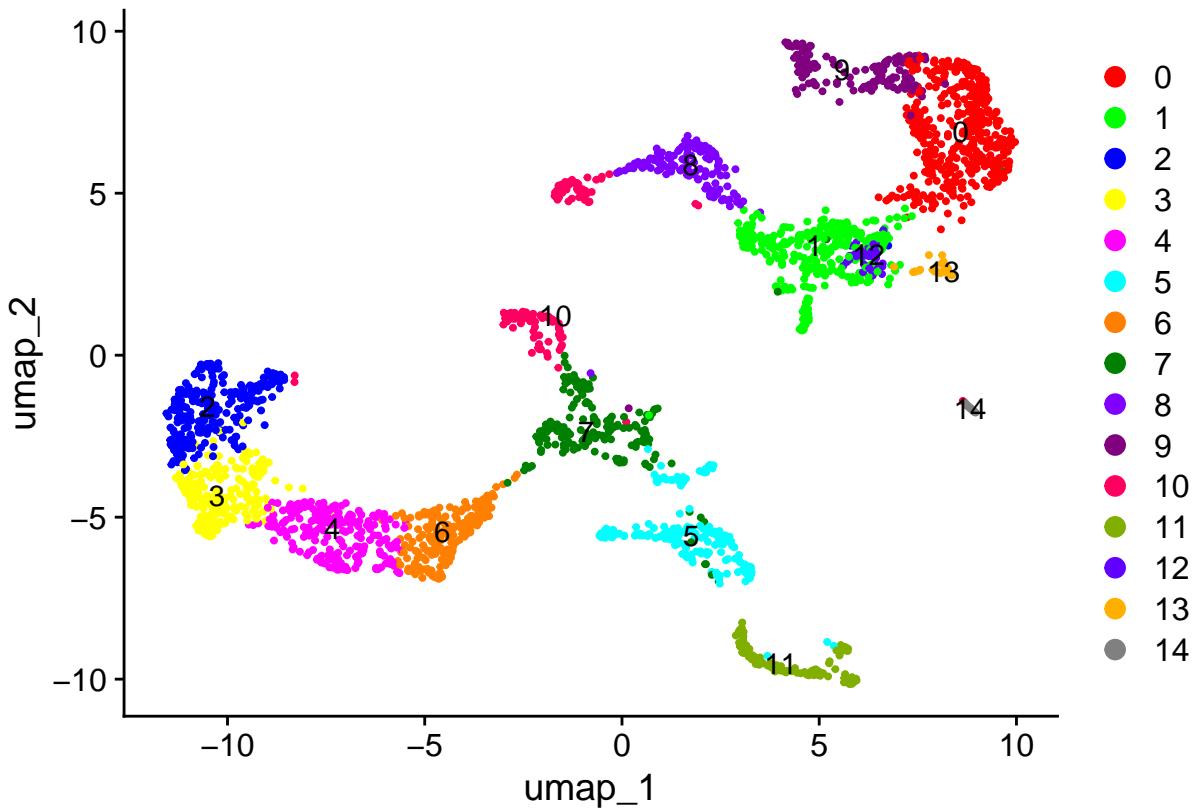
Some colors are similar and misleading, so we can replace them. We start with most vivid colors and we can adjust them interactively.

```
new_colors <- c(
  "#FF0000", "#00FF00", "#0000FF", "#FFFF00", "#FF00FF", "#00FFFF", "#FF8000",
  "#008000", "#8000FF", "#800080", "#FF0060", "#80AF00", "#6000FF", "#FFAF00",
  "#808080")
```

```
SpatialDimPlot(brain, label = TRUE, label.size = 3, pt.size.factor = 2.7) +
  scale_fill_manual(values=new_colors)
```

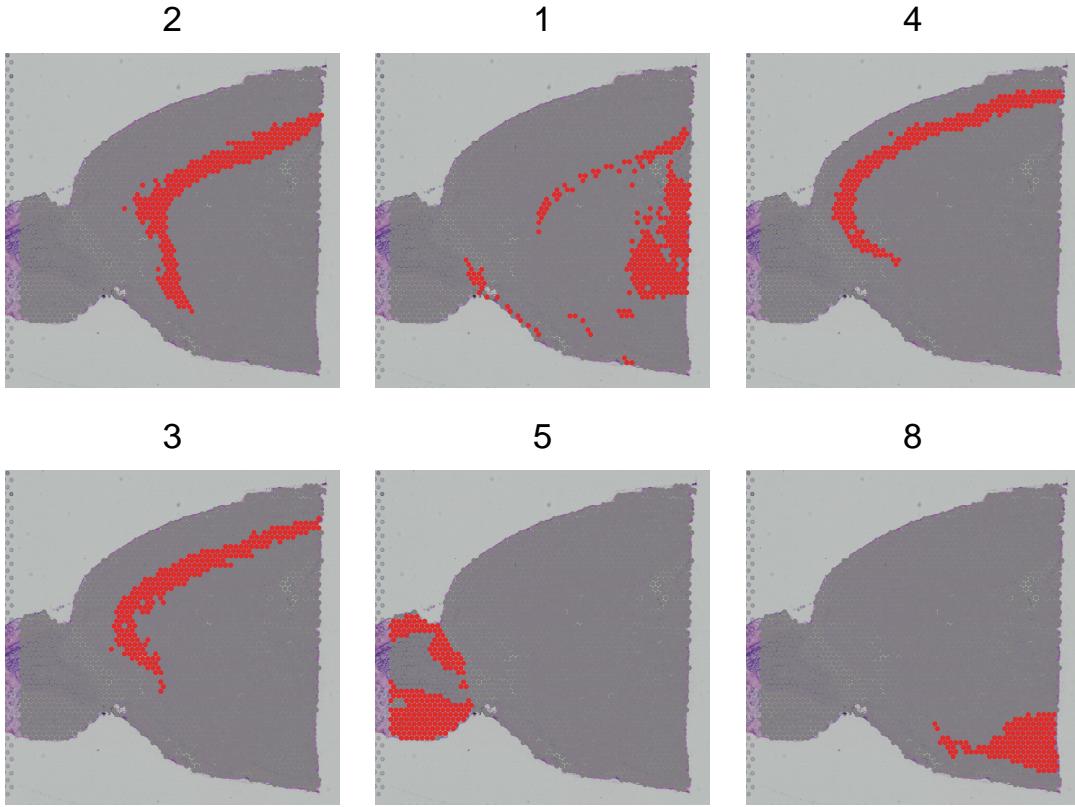


```
DimPlot(brain, reduction = "umap", label = TRUE) +  
  scale_color_manual(values=new_colors)
```



We can edit colors to our liking, but it may also help to look at clusters separately using `cell_highlight` parameter

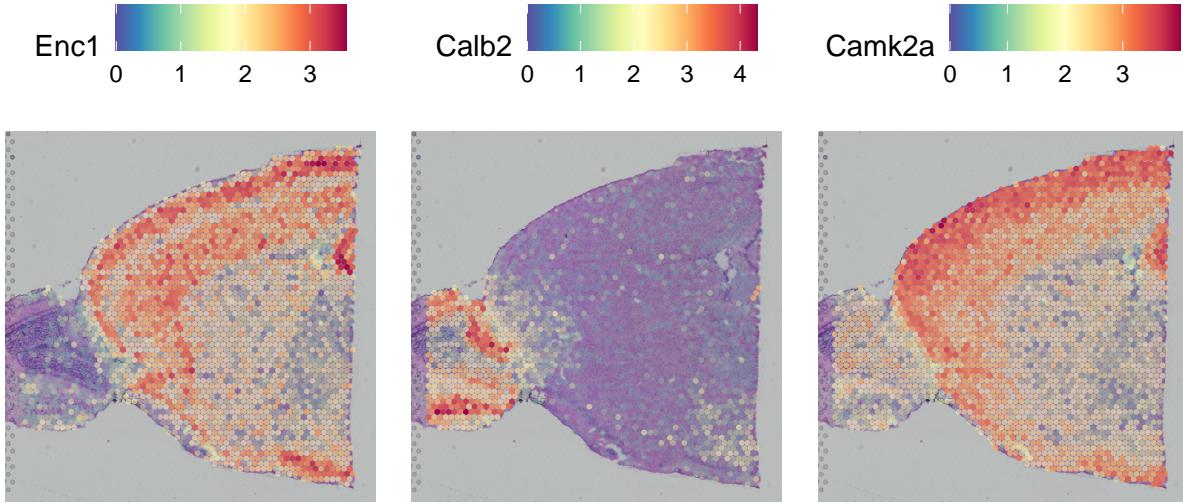
```
SpatialDimPlot(brain, cells.highlight = CellsByIdentities(object = brain,
  idents = c(2, 1, 4, 3, 5, 8)), facet.highlight = TRUE, ncol = 3,
  pt.size.factor = 2.7)
```



```
# Identification of Spatially Variable Features
```

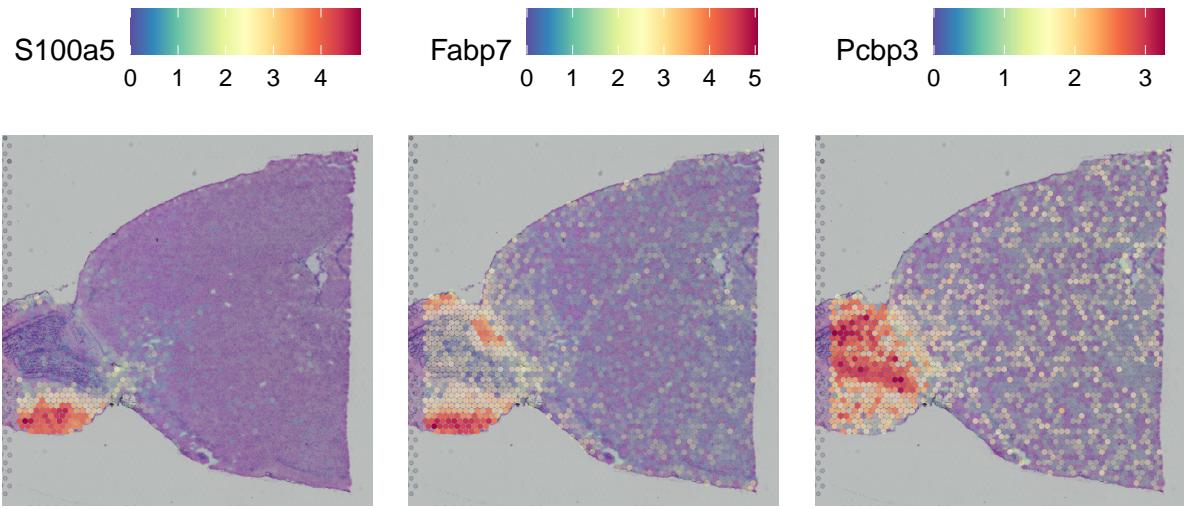
The first approach is the same as for single cell, find top markers of the clusters. Here are three top markers for cluster 5 versus cluster 6:

```
de_markers <- FindMarkers(brain, ident.1 = 5, ident.2 = 6)
SpatialFeaturePlot(object = brain, features = rownames(de_markers)[1:3],
pt.size.factor = 2.7, alpha = c(0.1, 1), ncol = 3)
```

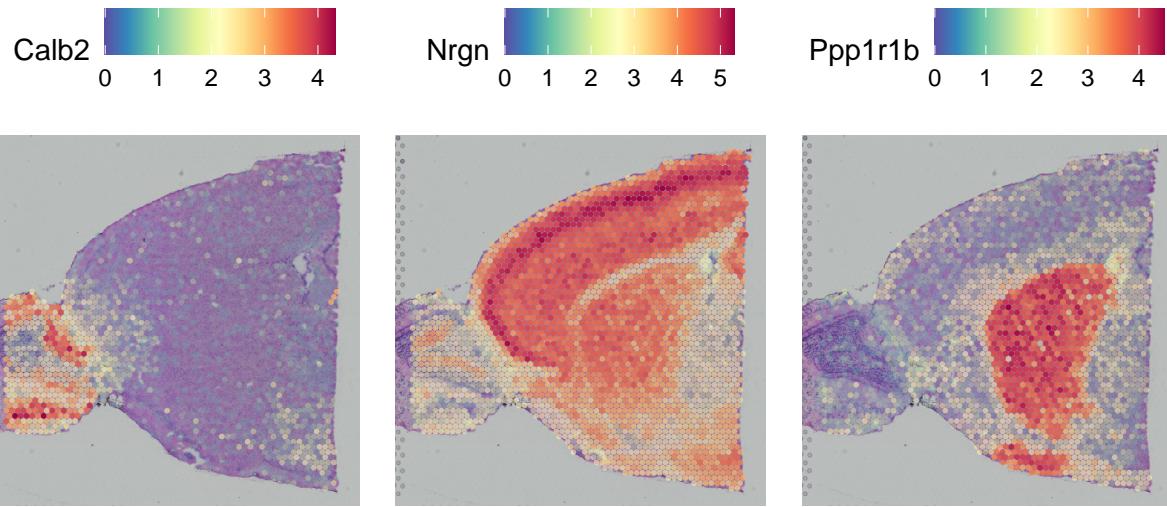


A more direct approach, `FindSpatiallyVariables()`. Those features are highly variable, but with the tendency that neighboring voxels have similar values. It must be stressed that this general concept can be implemented by multiple algorithms, but this one is worth trying. Like `FindMarkers`, it produces a ranked list of features/genes, and here we display top 9:

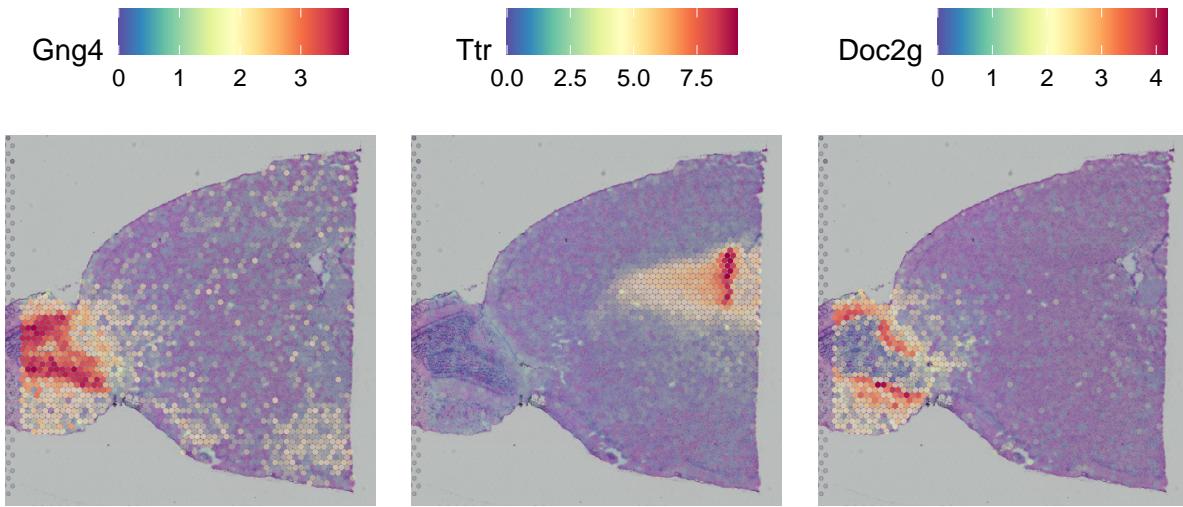
```
brain <- FindSpatiallyVariableFeatures(brain, assay = "SCT", features =
  VariableFeatures(brain)[1:1000], selection.method = "moransi")
# does not work
#top.features <- head(SpatiallyVariableFeatures(brain, selection.method =
#  "moransi"), 6)
top.features <- top_spatial_features <- brain@assays$SCT@meta.features %>%
  top_n(9, MoransI_observed) %>% row.names()
SpatialFeaturePlot(brain, features = top.features[1:3], ncol = 3,
  alpha = c(0.1, 1), pt.size.factor = 2.7)
```



```
SpatialFeaturePlot(brain, features = top.features[4:6], ncol = 3,  
alpha = c(0.1, 1), pt.size.factor = 2.7)
```

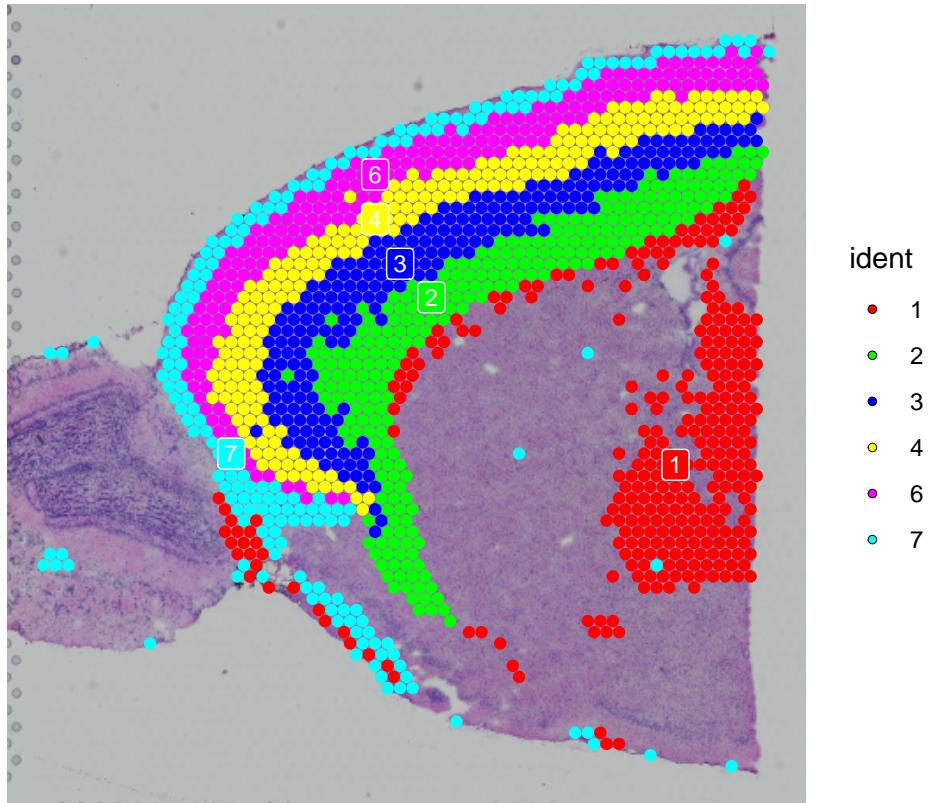


```
SpatialFeaturePlot(brain, features = top.features[7:9], ncol = 3,  
alpha = c(0.1, 1), pt.size.factor = 2.7)
```



We have seen the distribution of hippocampus marker Hpc and spacially correlated Ppp1r1b, so clusters 0 and 9 may show hippocampus. But where is cortex? We will return to this question later, but for now we assume that it consists of clusters with numbers in `c(1, 2, 3, 4, 6, 7)`

```
cortex <- subset(brain, idents = c(1, 2, 3, 4, 6, 7))
SpatialDimPlot(cortex, label = TRUE, pt.size.factor = 2.7, label.size = 3) +
  scale_fill_manual(values=new_colors)
```



We may want to crop a part of this plot that illustrates our important point, so this is an example how to do it. First, we need to know the coordinates used in the plot:

```
space_coords <- brain@images$anterior1@boundaries$centroids@coords
summary(space_coords[,1])
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	2685	5200	6757	6706	8194	10469

But then it becomes difficult, because the previous solution does not work, and there is no explicit data frame to create an input to plot or ggplot.