# Integration

## Material

https://youtu.be/2TsW5A53hTg?si=Bas__YskLl21-YjF

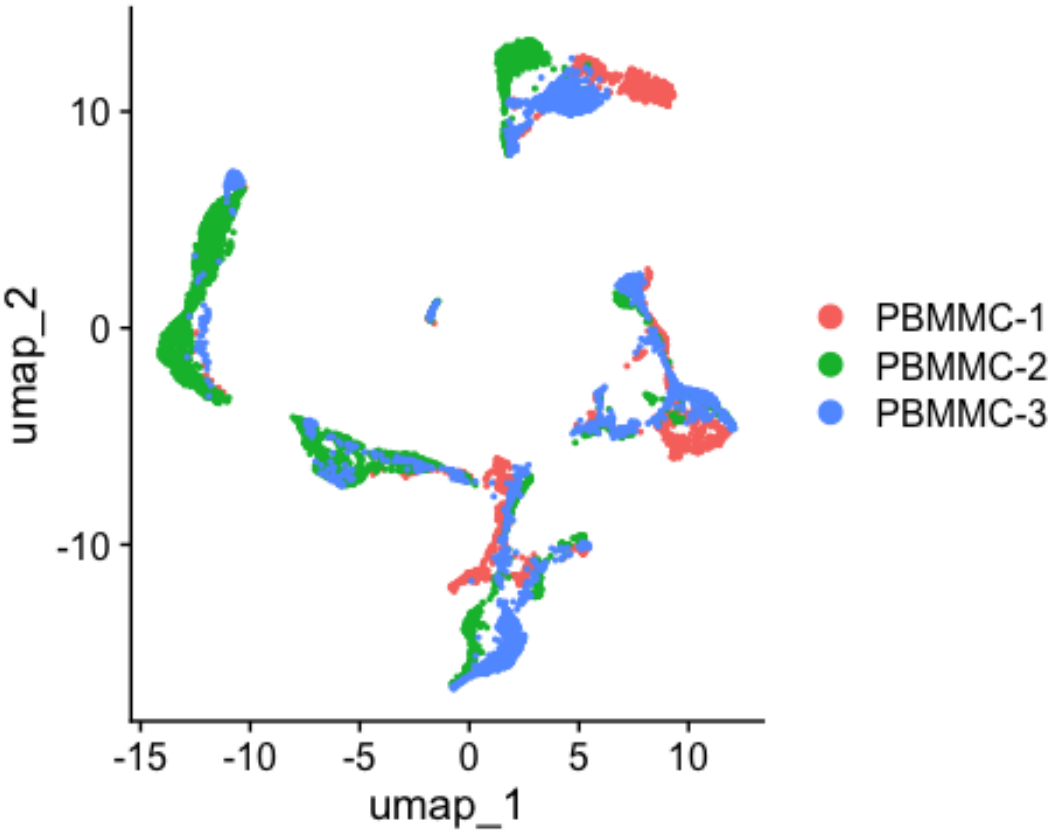
## Exercises

Let's have a look at the UMAP again. Although cells of different samples are shared amongst 'clusters', you can still see seperation within the clusters:

Code starts here:

```
Seurat::DimPlot(seu, reduction = "umap")
```

Code ends here




To perform the integration, we split our object by sample, resulting into a set of layers within the `RNA` assay. The layers are integrated and stored in the reduction slot - in our case we call it `integrated.cca`. Then, we re-join the layers

Code starts here:

```
seu[["RNA"]] <- split(seu[["RNA"]], f = seu$orig.ident)

seu <- Seurat::IntegrateLayers(object = seu, method = CCAIntegration,
                        orig.reduction = "pca",
                        new.reduction = "integrated.cca",
                        verbose = FALSE)

# re-join layers after integration
seu[["RNA"]] <- JoinLayers(seu[["RNA"]])
```

Code ends here

We can then use this new integrated matrix for clustering and visualization. Now, we can re-run and visualize the results with UMAP.

**Exercise**

Create the UMAP again on the `integrated.cca` reduction (using the function `RunUMAP` - set the option `reduction` accordingly). After that, generate the UMAP plot. Did the integration perform well?

**Answer**

Performing the scaling, PCA and UMAP:

Code starts here:

```
seu <- RunUMAP(seu, dims = 1:30, reduction = "integrated.cca")
```

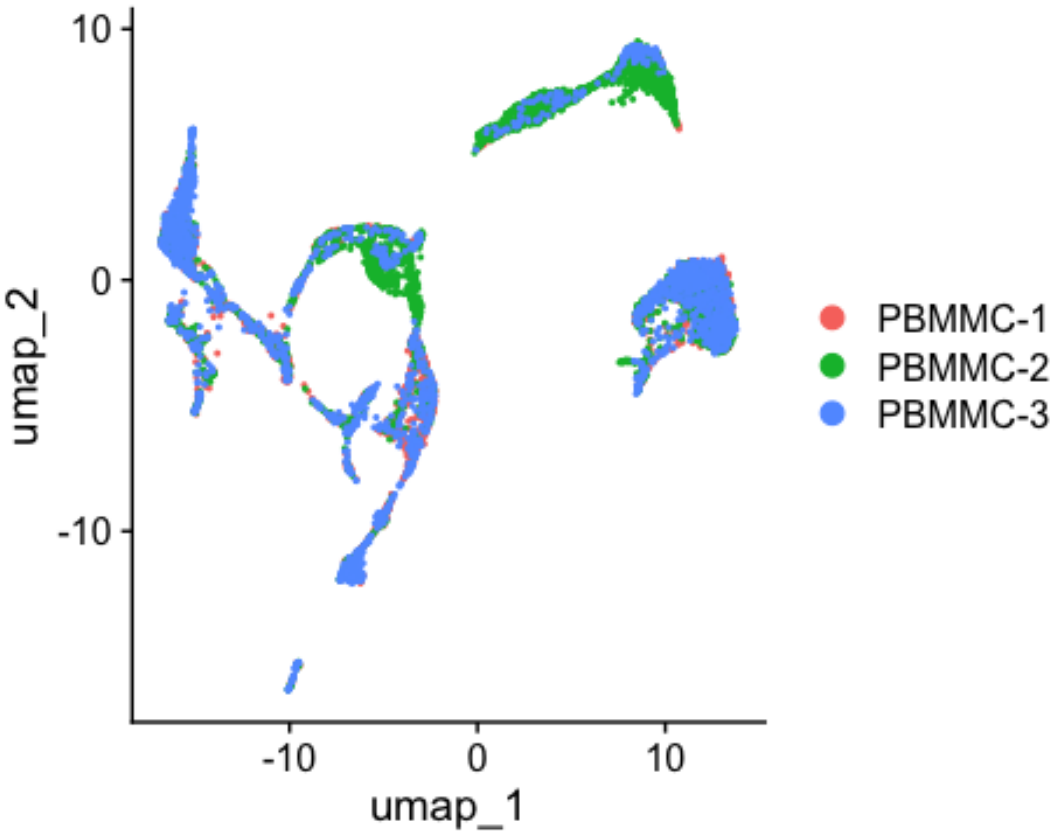
Code ends here

Plotting the UMAP:

Code starts here:

```
Seurat::DimPlot(seu, reduction = "umap")
```

Code ends here




## Save the dataset and clear environment

Code starts here:

```
saveRDS(seu, "day2/seu_day2-3.rds")
```

Code ends here

Clear your environment:

Code starts here:

```
rm(list = ls())
gc()
.rs.restartR()
```

Code ends here