External validation

Ricardo Martins-Ferreira

2023-08-01

External validation with the Olah et al. (2020)

```
libs <- c("Seurat", "tidyverse","fgsea",
          "org.Hs.eg.db","data.table",
          "ggplot2","ggalluvial",
          "magrittr","ggpubr",
          "RColorBrewer", "ComplexHeatmap",
          "circlize")
suppressMessages(
  suppressWarnings(sapply(libs, require, character.only = TRUE))
)
```

```
##         Seurat      tidyverse          fgsea   org.Hs.eg.db     data.table
##           TRUE           TRUE           TRUE           TRUE           TRUE
##        ggplot2     ggalluvial       magrittr         ggpubr   RColorBrewer
##           TRUE           TRUE           TRUE           TRUE           TRUE
## ComplexHeatmap       circlize
##           TRUE           TRUE
```

The raw counts matrix ("counts") and a corresponding cell annotation file ("annotation") were downloaded from https://github.com/vilasmenon/Microglia_Olah_et_al_2020

Create Seurat object and process it

```
setkeyv(counts, colnames(counts)[1])
rownames(counts) <- counts[[1]]
counts[[1]] <- NULL

Seurat = CreateSeuratObject(counts =counts, min.cells=3)
Seurat
```

```
## An object of class Seurat
## 20274 features across 16245 samples within 1 assay
## Active assay: RNA (20274 features, 0 variable features)
```

```
# Add annotation to metadata
metadata <- Seurat@meta.data

metadata$sample_id <- rownames(metadata)

metadata <- metadata %>% left_join(annotation, by="sample_id")
rownames(metadata)<- metadata$sample_id
```

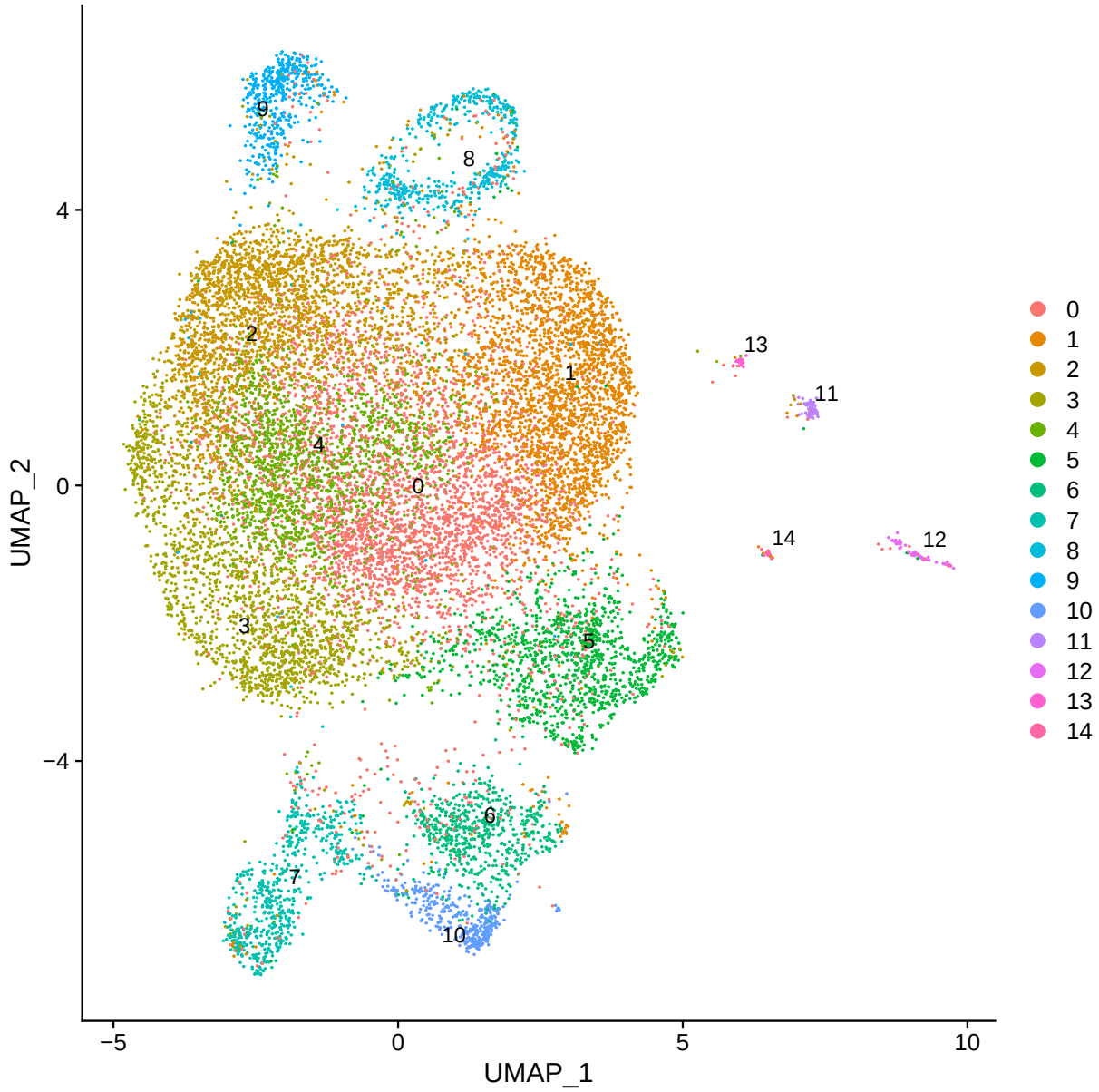
```
metadata -> Seurat@meta.data

# Normalization
## run sctransform
Seurat <- SCTransform(Seurat, verbose = FALSE,conserve.memory = TRUE,
                      vars.to.regress = "batch")
#defaul variable features = 3000
# include batch in vars.to.regress otherwise the final object will have major batch effect

# Perform linear dimensiona reduction
Seurat <- RunPCA(Seurat, verbose = FALSE)
```

Dimensionality reduction and clustering

```
Seurat <- RunUMAP(Seurat,  reduction = "pca",dims = 1:30, verbose = FALSE)
Seurat <- FindNeighbors(Seurat,  reduction = "pca",dims = 1:30)
Seurat <- FindClusters(Seurat,resolution = 0.5) #15 clusters
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 16245
## Number of edges: 513483
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8196
## Number of communities: 15
## Elapsed time: 3 seconds
```

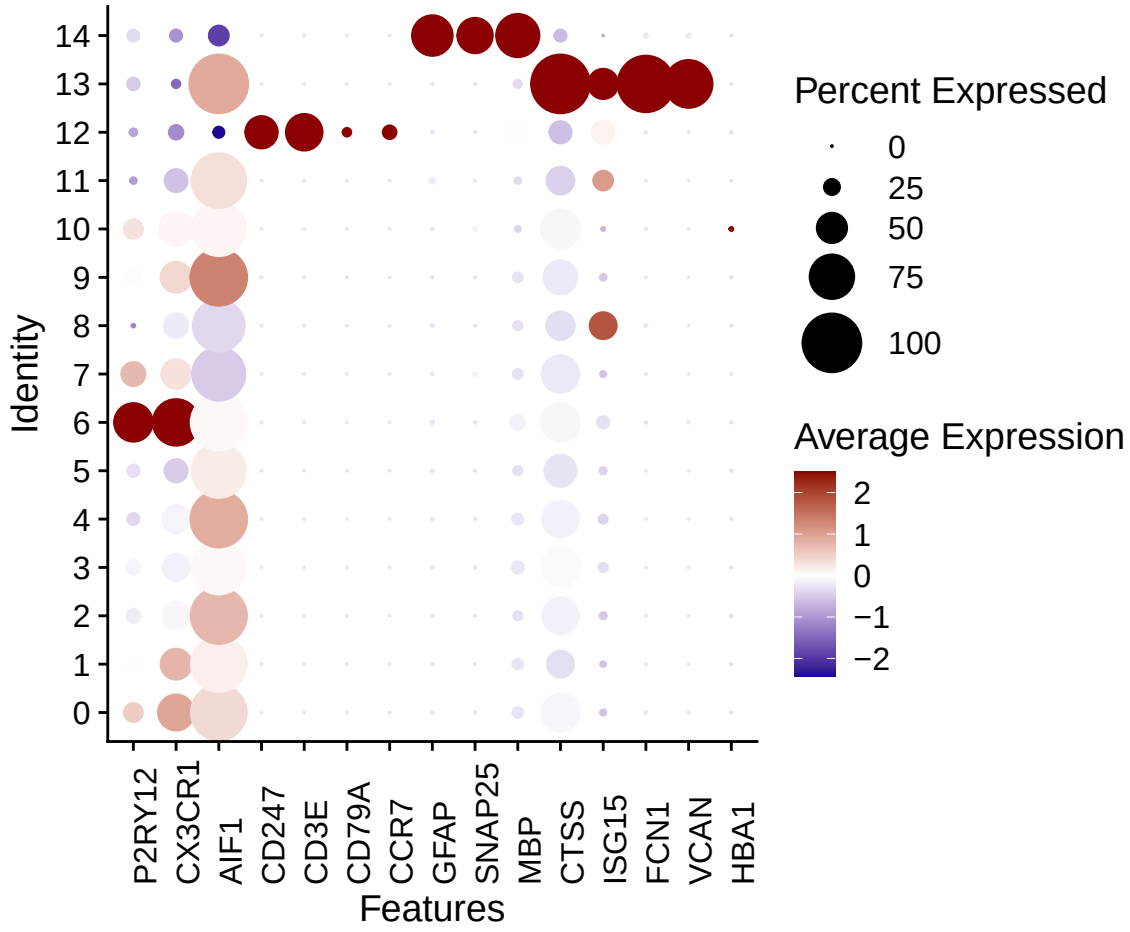
```
# Visualization
## UMAP
DimPlot(Seurat, reduction = "umap", label = TRUE, repel = TRUE)
```



Check for non-microglia clusters

```
DotPlot(Seurat, features = c("P2RY12","CX3CR1",
                             "AIF1", "CD247",
                             "CD3E", "CD79A",
                             "CCR7","GFAP",
                             "SNAP25", "MBP",
                             "CTSS","ISG15",
                             "FCN1","VCAN","HBA1"), dot.scale = 10,
        group.by ="seurat_clusters") +
  scale_colour_gradient2(low = "darkblue", mid = "white",
                         high = "darkred")+
  theme(axis.text.x = element_text(angle=90, hjust = 0))
```

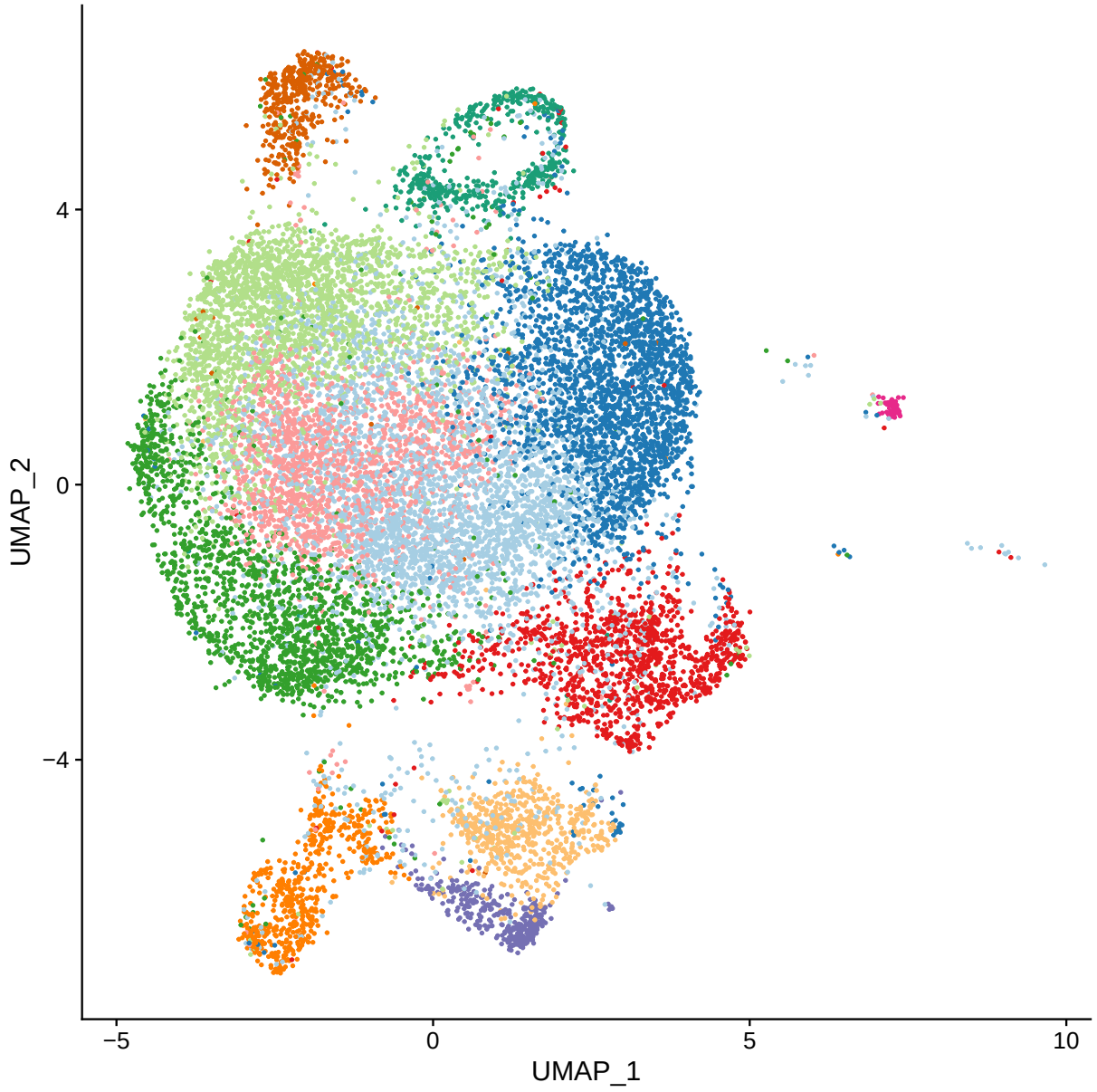
```
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```




```
# remove non-microglia clusters 15, 16, 17)
MG_Seurat <- subset(Seurat, idents=c("0","1","2","3","4","5",
                                     "6","7","8","9","10","11"))
```

Check for non-microglia clusters

```
coul <- c(brewer.pal(8, "Paired"),brewer.pal(7,"Dark2"))
DimPlot(MG_Seurat, reduction = "umap", label = F,
        repel = TRUE, cols = coul,pt.size = 0.5)+NoLegend()
```



Calculate DEGs to perform GSEA enrichment

```
Seurat.markers <- FindAllMarkers(MG_Seurat, only.pos = FALSE,
                                 min.pct = 0.1,
                                 logfc.threshold = 0.0,
                                 cores=16)
```

Perform GSEA enrichment of HuMicA genesets

Upload HuMicA genesets

The outputs from FindAllMarkers (min.pct = 0.1, logfc.threshold = 0.0) for the respective microglia clusters.

```
Homeos1_up$cluster <- "Homeos1"
Homeos2_up$cluster <- "Homeos2"
Homeos3_up$cluster <- "Homeos31"
Homeos4_up$cluster <- "Homeos4"
DIM_up$cluster <- "DIM"
Intermediate.DAM_up$cluster <- "Intermediate.DAM"
Final.DAM_up$cluster <- "Final.DAM"

Atlas_cluster <- rbind(Homeos1_up,Homeos2_up,Homeos3_up,Homeos4_up,
                       DIM_up, Intermediate.DAM_up, Final.DAM_up)

colnames(Atlas_cluster) <- c("Gene","cluster")

Atlas_cluster <- split(x=Atlas_cluster$Gene, f=Atlas_cluster$cluster )
```

fgsea

```
# Subset Seurat.Markers results from Olah in cluster into a list
clusterlist <- c("0","1","2","3","4","5","6","7","8","9","10","11")

p <- list()
results <- data.frame()
for (i in 1:length(clusterlist)) {
  print(i)
  genes <- Seurat.markers[Seurat.markers$cluster==clusterlist[i],]
  genes<- structure(genes$avg_log2FC, names=genes$gene)
  genes <- fgsea(pathways =Atlas_cluster,
                 stats    = genes,
                 scoreType ="pos",
                 minSize  = 0,
                 maxSize  = Inf)
  genes <- as.data.frame(apply(genes, 2, as.character))

  genes<- as.data.frame(genes) %>% mutate(cluster = clusterlist[i])
  results <- rbind(results,genes)
  p[[i]]<- genes

}
```

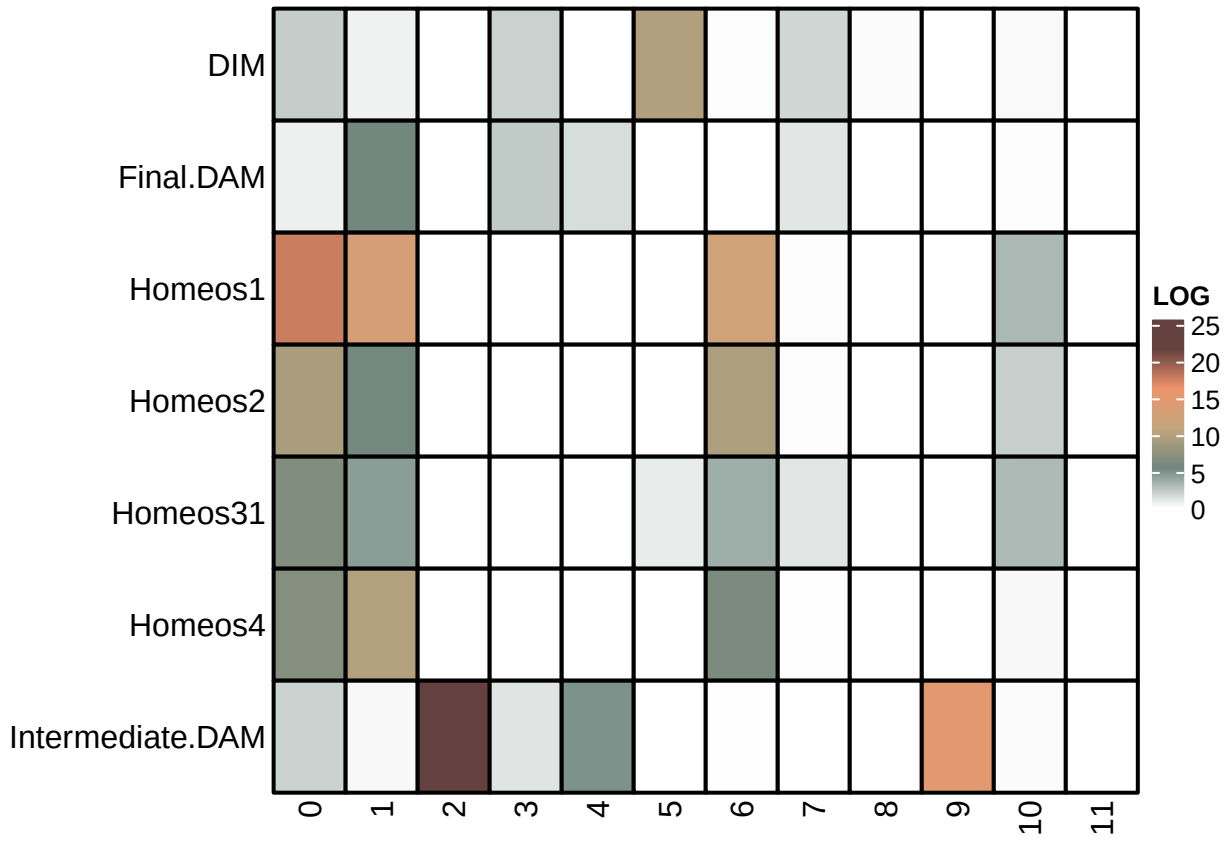
```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
```

```
# Heatmap with - log adj p value
rownames(results) <- NULL
```

```
results$padj<- as.numeric(results$padj)
results$LOG <- -log10(results$padj)

##Heatmap of fgsea results
clusterlist2 <- unique(results$cluster)
mat<- data.frame(pathway=unique(results$pathway))
for (i in 1:length(clusterlist2)) {
  print(i)
  df <- results[results$cluster==clusterlist2[i],]
  #rownames(df)<-df$pathway
  df <- df[,c("pathway","LOG")]
  colnames(df)[2]<-clusterlist2[i]
  mat <- left_join(mat,df,by="pathway")
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
```

```
rownames(mat)<- mat$pathway
mat[is.na(mat)] <- 0
mat<-mat[,-1]
mat<- mat %>% mutate_if(is.character, as.numeric)
mat<- as.matrix(mat)


buylrd <- c("white", "#70877F", "#C4A77D",
            "#EF946C", "#64403E")
colors.martin <- colorRampPalette(buylrd)(100)


Heatmap(mat,col = colors.martin,show_column_dend = F,
        border_gp = gpar(col = "black", lwd = 1),
        rect_gp = gpar(col = "black", lwd = 2), cluster_rows = F,
        cluster_columns=F,show_row_names = TRUE,show_column_names = TRUE,
        name = "LOG",row_names_side = "left",na_col = "grey")
```



Integrative analysis of the HuMicA and Olah's single-cell data with Transfer label

The Atlas object represents the original HuMicA with 64,438 nuclei. The subset_Atlas object includes only the clusters of interest.

```
subset_Atlas <- subset(Atlas, idents=c("0","1","2","4","5","10","11"))
```

Find anchors between datasets and selecting genes common in both

```
# Both objects to be integrated need have the "SCT" assay active

## HuMicA
DefaultAssay(subset_Atlas) <- "SCT"

# Olah's microglial clusters
DefaultAssay(MG_Seurat) <- "SCT"

features <- intersect(rownames(subset_Atlas@assays[["SCT"]]@counts),
                      rownames(MG_Seurat@assays[["SCT"]]@counts))

Atlas_Olah_anchors <- FindTransferAnchors(reference = subset_Atlas,
                                          query =MG_Seurat,
                                          dims = 1:30,
                                          normalization.method = "SCT",
```

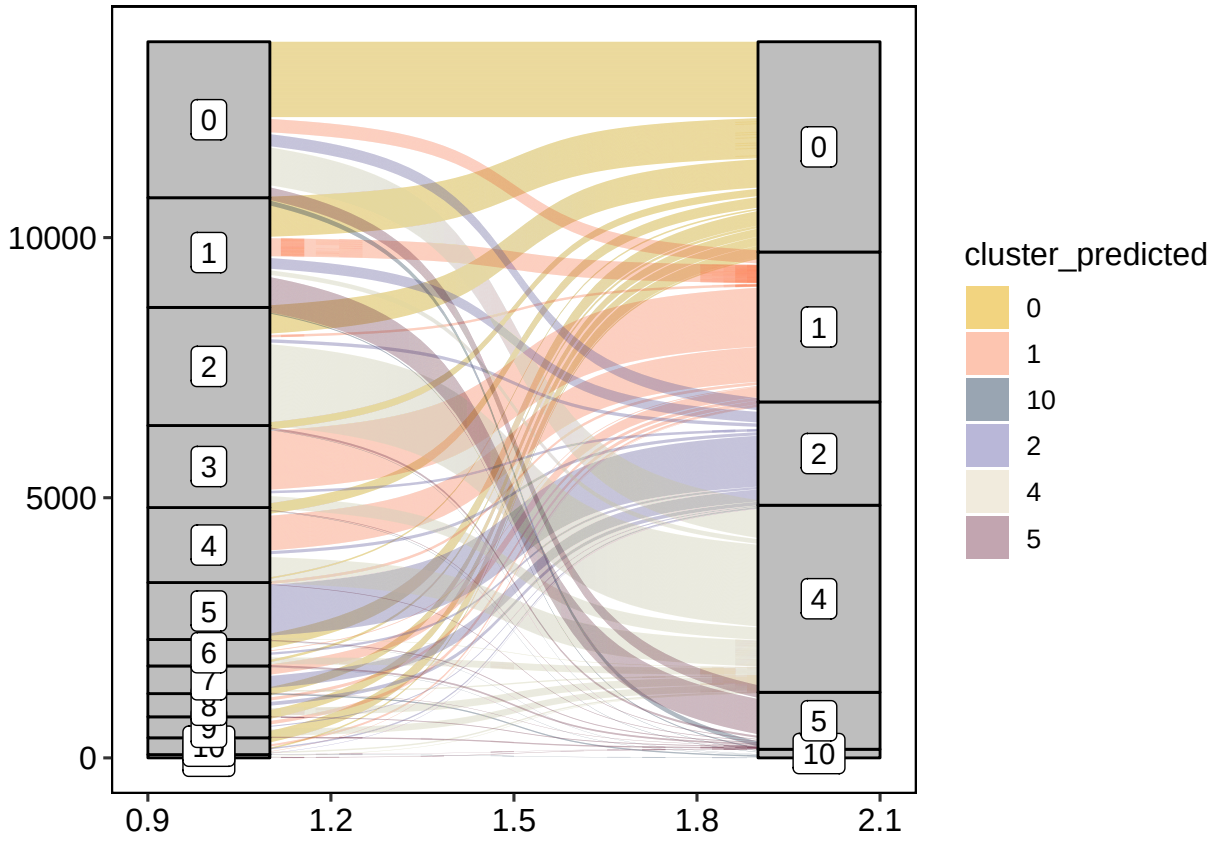
```
                                          recompute.residuals = FALSE,
                                          reference.reduction = "pca",
                                          features = features)

predictions <- TransferData(anchorset = Atlas_Olah_anchors,
                            refdata = subset_Atlas$integrated_snn_res.0.25,
                            dims = 1:30)
MG_Seurat <- AddMetaData(MG_Seurat, metadata = predictions)
# Adding the column of predicted ID to Atlas metadata's
```

Sanley plot

```
plot <- data.frame(cluster_original = MG_Seurat$SCT_snn_res.0.5,
                   cluster_predicted = MG_Seurat$predicted.id,
                   cluster_prob = MG_Seurat$prediction.score.max)%>%
  dplyr::filter(cluster_prob > 0.5)

is_alluvia_form(as.data.frame(plot), axes = 1:3, silent = TRUE)
```

```
## [1] TRUE
```

```
color3 <-c("#E6AB02", "#FC8D62","#344D67", "#7570B3" ,
           "#E5D8BD" ,"#874C62" , "#B3CDE3")

coul <- brewer.pal(12, "Paired")

ggplot(as.data.frame(plot),
       aes(axis1 = cluster_original, axis2 = cluster_predicted)) +
  geom_alluvium(aes(fill = cluster_predicted),
                curve_type = "arctangent",
                width = 1/12) +
  scale_fill_manual(values = color3 )+
  geom_stratum(width = 1/5, fill = "gray", color = "black") +
  geom_label(stat = "stratum", aes(label = after_stat(stratum))) +
  theme_pubr(border = TRUE, legend = 'right')
```

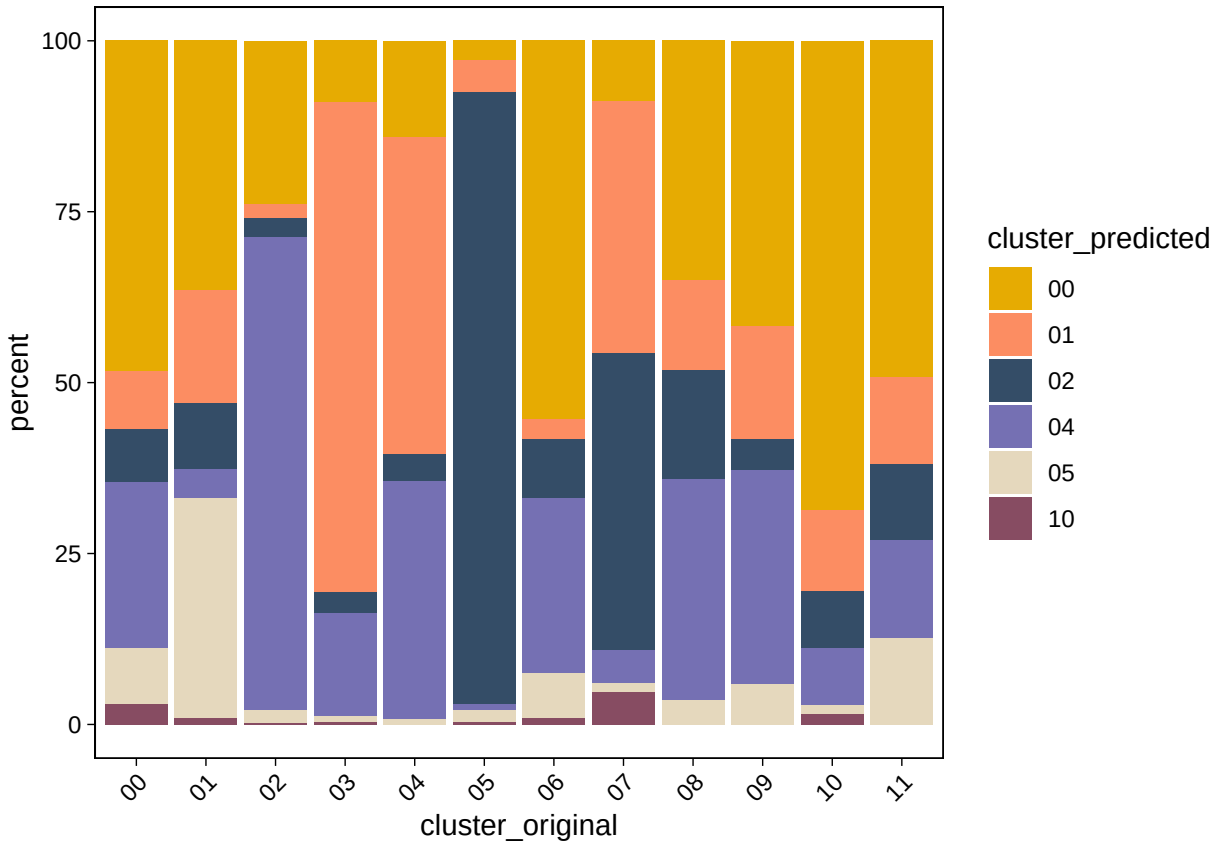


Stacked bar plot

Representation of the percentage of cells from each Olah cluster that show the highest prediction to the respective HuMicA clusters.

```
# Calculate proportions of clusters predicted by atlas cluster
data <- plot %>% group_by(cluster_original, cluster_predicted) %>%
  dplyr::summarise(Nb = n()) %>%
  dplyr::mutate(C = sum(Nb))%>%
  dplyr::mutate(percent = Nb/C*100)

data$percent2 <- format(round(data$percent,2), nsmall=2)


## Stacked Bar plot per Group
data <-data[order(as.numeric(as.character(data$cluster_original))), ]
data$cluster_predicted <- as.factor(data$cluster_predicted)




#replace cluster 1 by cluster 01, etc...
data <- data %>% mutate(cluster_original=str_replace(cluster_original,"^0$","00"))%>%
  mutate(cluster_original = str_replace(cluster_original, "^1$","01")) %>%
  mutate(cluster_original = str_replace(cluster_original, "^2$","02")) %>%
  mutate(cluster_original = str_replace(cluster_original, "^3$","03")) %>%
```

```
  mutate(cluster_original = str_replace(cluster_original, "^4$","04")) %>%
  mutate(cluster_original = str_replace(cluster_original, "^5$","05")) %>%
  mutate(cluster_original = str_replace(cluster_original, "^6$","06")) %>%
  mutate(cluster_original = str_replace(cluster_original, "^7$","07")) %>%
  mutate(cluster_original = str_replace(cluster_original, "^8$","08")) %>%
  mutate(cluster_original = str_replace(cluster_original, "^9$","09"))


data <- data %>% mutate(cluster_predicted=str_replace(cluster_predicted,"^0$","00"))%>%
  mutate(cluster_predicted = str_replace(cluster_predicted, "^1$","01")) %>%
  mutate(cluster_predicted = str_replace(cluster_predicted, "^2$","02")) %>%
  mutate(cluster_predicted = str_replace(cluster_predicted, "^3$","03")) %>%
  mutate(cluster_predicted = str_replace(cluster_predicted, "^4$","04")) %>%
  mutate(cluster_predicted = str_replace(cluster_predicted, "^5$","05")) %>%
  mutate(cluster_predicted = str_replace(cluster_predicted, "^6$","06")) %>%
  mutate(cluster_predicted = str_replace(cluster_predicted, "^7$","07")) %>%
  mutate(cluster_predicted = str_replace(cluster_predicted, "^8$","08")) %>%
  mutate(cluster_predicted = str_replace(cluster_predicted, "^9$","09"))

ggplot(data, aes(x = cluster_original,
                 y = percent, fill = cluster_predicted))+
  geom_bar(stat = "identity",)+
  scale_fill_manual(values=color3)+
  theme_linedraw()+
  theme(panel.grid=element_blank(),
        axis.text.x = element_text(angle = 45, hjust = 1))
```


100
75
50
25
0
percent
00
01
02
03
04
05
06
07
08
09
10
11
cluster_original
cluster_predicted
00
01
02
04
05
10