

Cluster distribution

Ricardo Martins-Ferreira

2023-07-27

Distribution of nuclei populations or clusters across different pathologies

```
libs <- c("Seurat", "tidyverse", "gridExtra", "sccomp", "ggplot2", "ggpubr")
suppressMessages(
  suppressWarnings(sapply(libs, require, character.only = TRUE))
)
```

```
##      Seurat tidyverse gridExtra      sccomp      ggplot2      ggpubr
##      TRUE         TRUE         TRUE         TRUE         TRUE         TRUE
```

Upload Seurat object

The Seurat object consists of the 64,438 nuclei already clustered and annotated for clinicopathological features of the subjects. This object is divided in 15 clusters.

Calculate the proportions of clusters per pathology

```
metadata <- Seurat@meta.data

# percentage of cluster in each pathology ("Group")
data <- metadata[,c("Subject", "Group", "integrated_snn_res.0.25")]
colnames(data) <- c("Subject", "Group", "Cluster")

data <- data %>% group_by(Group, Cluster) %>%
  dplyr::summarise(Nb = n()) %>%
  dplyr::mutate(C = sum(Nb)) %>%
  dplyr::mutate(percent = Nb/C*100)

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

#percentage of cluster in the whole dataset
metadata$whole <- "ALL"
data2 <- metadata[,c("whole", "integrated_snn_res.0.25")]
colnames(data2) <- c("whole", "Cluster")

data2 <- data2 %>% group_by(whole, Cluster) %>%
  dplyr::summarise(Nb = n()) %>%
  dplyr::mutate(C = sum(Nb)) %>%
  dplyr::mutate(percent = Nb/C*100)
```

```
data2$percent2 <- format(round(data2$percent,2), nsmall=2)
colnames(data2)[1]<- "Group"

data <- rbind(data,data2)
```

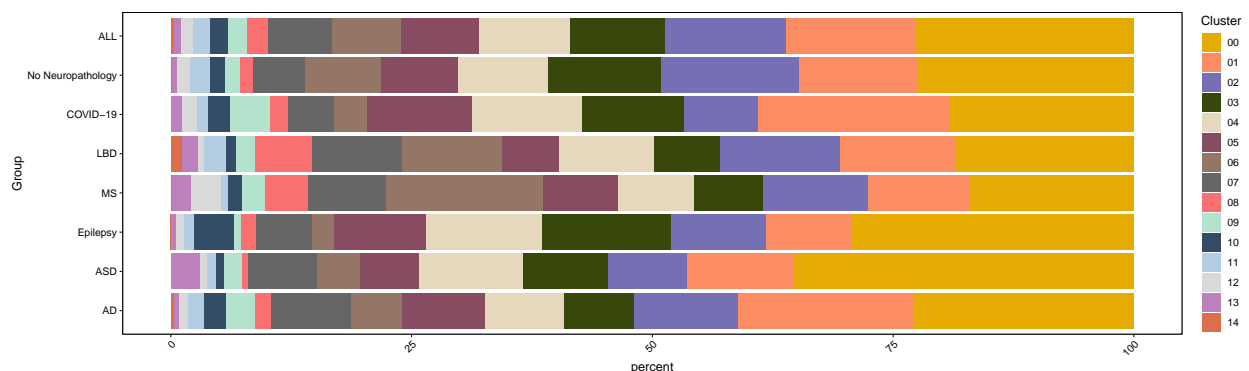
Stacked Bar plot per “Group”

```
color_clusters <-c("#E6AB02", "#FC8D62", "#7570B3", "#38470B", "#E5D8BD", "#874C62",
                  "#937666", "#666666", "#FA7070", "#B3E2CD", "#344D67", "#B3CDE3",
                  "#D9D9D9", "#BC80BD", "#DE6E4B", "#430D27")

#replace cluster 1 by cluster 01, etc...

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

ggplot(data, aes(x = Group,
                 y = percent, fill = Cluster))+
  scale_x_discrete(limits=c("AD","ASD","Epilepsy","MS","LBD", "COVID-19",
                           "No Neuropathology","ALL"))+
  geom_bar(stat = "identity",)+
  scale_fill_manual(values=color_clusters)+
  theme_linedraw()+
  coord_flip()+
  theme(panel.grid=element_blank(), axis.text.x = element_text(angle = 45, hjust = 1))
```



Calculate the proportions of clusters per subject

```
metadata <- Seurat@meta.data
data <- metadata[,c("Subject","Group","Study","integrated_snn_res.0.25")]
```

```
colnames(data)<- c("Subject","Group","Study","Cluster")

data <- data %>% group_by(Subject, Group,Study,Cluster) %>%
  dplyr::summarise(Nb = n()) %>%
  mutate(C = sum(Nb))%>%
  mutate(percent = Nb/C*100)
```

Stacked Bar plot per “Subject” (grouped by “Group”)

```
# loop for each cluster
cluster_list <- c("0","1","2","3","4","5","6","7","8","9","10","11","12","13","14")

group_colors <- c("#A4243B", "#D8973C", "#BD632F", "#4C956C", "#273E47", "#720026", "#81C3D7")

p<-list()

for (i in 1:length(cluster_list)) {

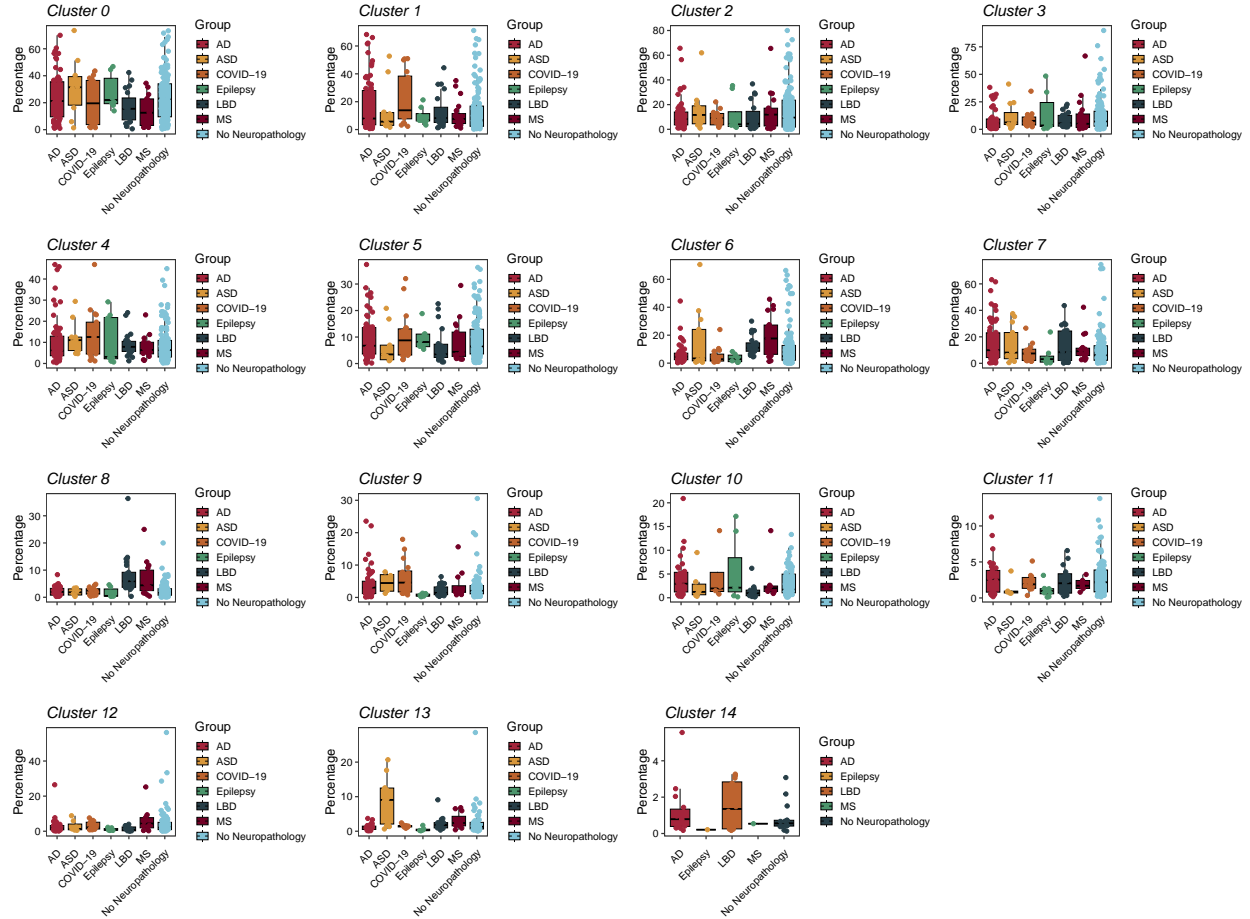
  data_cluster <- data[data$Cluster==cluster_list[i],]

  p[[i]]<- ggplot(data_cluster, aes(y=percent, x=as.factor(Group),fill=Group , color = Group)) +
    geom_boxplot(color="black",outlier.shape =NA)+
    geom_point(aes(colour=Group), size=2,
               position = position_jitterdodge(jitter.width = NULL,
                                               jitter.height = 0,dodge.width = 0.75,
                                               seed=NA)) +

    scale_fill_manual("Group",values=group_colors)+
    scale_color_manual("Group",values=group_colors)+
    theme_bw(base_size=15) +
    ylab('Percentage') +
    labs(title = paste0("Cluster ",cluster_list[i]))+
    xlab('') +
    theme(axis.text.x = element_text(angle = 45, vjust = 0.9,
                                     hjust = 1, colour = "black"),
          axis.text.y = element_text(colour = "black"),
          panel.grid.minor = element_blank(),
          panel.grid.major = element_blank(),
          plot.title = element_text(hjust=0, face = "italic"))

}

do.call(grid.arrange, p)
```



Calculate the significance of differential distribution with sccomp

The significance is calculated for each pathology (“Group”) in comparison to the control population (“Healthy” or “No Neuropathology”)

```
colnames(Seurat@meta.data)[16] <- "ReplicateID"

Seurat@meta.data <- Seurat@meta.data %>%
  mutated(Group = str_replace(Group, "No Neuropathology", "Healthy"))

Seurat@meta.data <- Seurat@meta.data %>%
  mutate(Group = str_replace(Group, "COVID-19", "COVID19"))

####
#with contrasts
res = Seurat|>
  sccomp_glm(
    formula_composition = ~ 0+Group,
    contrasts = c("GroupAD - GroupHealthy", "GroupASD - GroupHealthy",
                  "GroupCOVID19 - GroupHealthy", "GroupEpilepsy - GroupHealthy",
                  "GroupLBD - GroupHealthy", "GroupMS - GroupHealthy"),
    .sample = ReplicateID,
    .cell_group = integrated_snn_res.0.25,
    bimodal_mean_variability_association = TRUE,
```

```
cores = 5
)
```

Plot sccomp output

```
## Rows: 90 Columns: 18
## -- Column specification -----
## Delimiter: ","
## chr  (2): parameter, factor
## dbl  (6): integrated_snn_res.0.25, c_lower, c_effect, c_upper, c_pH0, c_FDR
## lgl  (10): c_n_eff, c_R_k_hat, v_lower, v_effect, v_upper, v_pH0, v_FDR, v_n...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
res$sig<- ifelse(res$c_FDR<0.05, "yes", "no")

ggplot(res, aes(x = integrated_snn_res.0.25, y = res$c_effect, color=sig)) +
  geom_point(stat = "identity", shape = 15) +
  scale_color_manual(values = c("grey", "#A21F16"))+
  geom_hline(yintercept = c(-0.2,0.2), linetype = "dashed", color = "grey") +
  geom_errorbar(aes(ymin = c_lower, ymax = c_upper), width = 0.4) +
  facet_wrap(~parameter,scales="free")+
  ylab("Credible interval of the slope")+
  xlab("Cluster")+
  coord_flip()+
  theme_pubr()+
  border()
```

