ID Assignment and Cell Number

Jaun Ramiro Lorenzo

April 26, 2020

```
library(ggplot2)
library(sjmisc)
library(readr)
```

```
cured_brain_atlas_PL_and_OH_CircularOtherMarkers <- read_delim("../Datasets/cured brain atlas PL and OH
    ";", escape_double = FALSE, trim_ws = TRUE)
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   X1 = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
load("../2-PlotBestFirstClustering/CeNeurons_Seurat_object_1st.rda")  #seurat object
load("IterationData.rda")
IterationData <- unique(IterationData)
# Clusters <- list('4' = c('PVM','AVM'), '13' = c('CAN'),
# '14' = c('AVK'), '16' = c('FLP'), '20' = c('PLN','ALN'),
# '30' = c('RIA'), '31' = c('ADL'), '33' = c('AVL', 'DVB'),
# '34' = c('ASG'), '36' = c('AQR.PQR', 'URX'), '38' =
# c('ASJ'),'39' = c('RMG'), '40' = c('ASK'), '41' = c('DVA'),
# '43' = c('AVB'), '44'= c('BAG'), '45'= c('AIN'), '46' =
# c('IL2'), '48' = c('RMH'), '49' = c('AWA'), '50' =
# c('ADF'), '51'= c('ASEL'), '52' = c('AVG'), '53' =
# c('RIC'), '54'= c('AIA'), '55' = c('PVQ'), '59' = c('RIS'),
# '60' = c('M1'), '61' = c('ASI'), '62' = c('ASER'), '63' =
# c('I5'))

Clusters <- list(`13` = c("CAN"), `14` = c("AVK"), `16` = c("FLP"),
    `30` = c("RIA"), `31` = c("ADL"), `34` = c("ASG"), `38` = c("ASJ"),
    `39` = c("RMG"), `40` = c("ASK"), `41` = c("DVA"), `44` = c("BAG"),
    `45` = c("AIN"), `46` = c("IL2"), `49` = c("AWA"), `52` = c("AVG"),
    `53` = c("RIC"), `55` = c("PVQ"), `59` = c("RIS"), `60` = c("M1"),
    `62` = c("ASER"), `63` = c("I5"))



IterationData <- IterationData[IterationData$parent_cluster %in%
    names(Clusters), ]
IterationData <- cbind(IterationData, Good_assignment = NA)
```

```
IterationData <- cbind(IterationData, Perfect_assignment = NA)

sum_stats <- data.frame()
sum_stats_perfect <- data.frame()
# Check assignment
for (Cluster in names(Clusters)) {

    IterationData[IterationData$parent_cluster == Cluster, ]$Good_assignment <- unlist(lapply(IterationD
        Cluster, ]$cluster_id, FUN = str_contains, pattern = Clusters[[Cluster]],
        logic = "AND"))

    IterationData[IterationData$parent_cluster == Cluster, ]$Perfect_assignment <- IterationData[Iterati
        Cluster, ]$n_ident == length(Clusters[[Cluster]]) & IterationData[IterationData$parent_cluster =
        Cluster, ]$Good_assignment

    length(Clusters[[Cluster]])

    ClusterData <- IterationData[IterationData$parent_cluster ==
        Cluster, ]

    # save statistics
    sum_stats <- rbind(sum_stats, data.frame(c(Cluster = Cluster,
        Good_assignment = T, as.list(summary(ClusterData[ClusterData$Good_assignment ==
            T, c("n_cells")])))))

    sum_stats <- rbind(sum_stats, data.frame(c(Cluster = Cluster,
        Good_assignment = F, as.list(summary(ClusterData[ClusterData$Good_assignment ==
            F, c("n_cells")])))))

    sum_stats_perfect <- rbind(sum_stats_perfect, data.frame(c(Cluster = Cluster,
        Perfect_assignment = T, as.list(summary(ClusterData[ClusterData$Perfect_assignment ==
            T, c("n_cells")])))))

    sum_stats_perfect <- rbind(sum_stats_perfect, data.frame(c(Cluster = Cluster,
        Perfect_assignment = F, as.list(summary(ClusterData[ClusterData$Perfect_assignment ==
            F, c("n_cells")])))))

}
```

Summary statistics (Mean values for all clusters) - Iteration data from second screening where small clusters are obtained by increasing resolution

Number of cells for sub-clusters when the automatically assigned identities include the expected ones

```
#-Properly assigned:
apply(sum_stats[sum_stats$Good_assignment == T, c("Min.", "X1st.Qu.",
    "Median", "Mean", "X3rd.Qu.", "Max.")], 2, mean)
```

```
##      Min.  X1st.Qu.    Median      Mean  X3rd.Qu.      Max.
##  3.142857 12.535714 24.714286 28.649422 41.130952 85.857143
```

```
#-Badly assigned:
apply(sum_stats[sum_stats$Good_assignment == F, c("Min.", "X1st.Qu.",
    "Median", "Mean", "X3rd.Qu.", "Max.")], 2, mean)
```

```
##      Min.  X1st.Qu.    Median      Mean  X3rd.Qu.      Max.
##  2.000000  2.523810  3.619048  4.673302  5.571429 21.333333
```

Number of cells for sub-clusters when the automatically assigned identities match exactly the expected

```
# Two clusters do not contain perfect matches
sum_stats_perfect <- sum_stats_perfect[sum_stats_perfect$Cluster !=
    "43", ]
sum_stats_perfect <- sum_stats_perfect[sum_stats_perfect$Cluster !=
    "48", ]
#-Properly assigned:
apply(sum_stats_perfect[sum_stats_perfect$Perfect_assignment ==
    T, c("Min.", "X1st.Qu.", "Median", "Mean", "X3rd.Qu.", "Max.")],
    2, mean)
```

```
##      Min.  X1st.Qu.    Median      Mean  X3rd.Qu.      Max.
##  3.571429 16.178571 33.690476 34.164162 47.809524 85.857143
```

```
#-Badly assigned:
apply(sum_stats_perfect[sum_stats_perfect$Perfect_assignment ==
    F, c("Min.", "X1st.Qu.", "Median", "Mean", "X3rd.Qu.", "Max.")],
    2, mean)
```

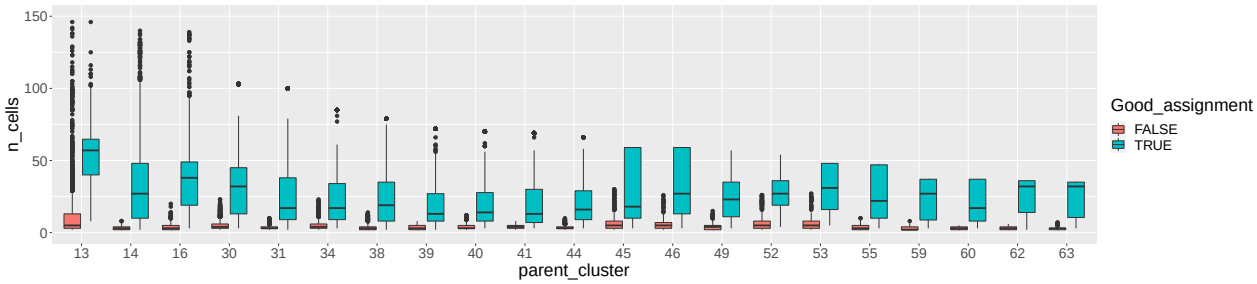
```
##      Min.  X1st.Qu.    Median      Mean  X3rd.Qu.      Max.
##  2.000000  2.809524  4.238095  6.077753  7.238095 34.333333
```

Boxplots

Number of cells for sub-clusters when the automatically assigned identities include the expected ones

```
# Box plot
p <- ggplot(IterationData, aes(x = parent_cluster, y = n_cells,
    fill = Good_assignment)) + ylim(0, 150) + theme(text = element_text(size = 20)) +
    geom_boxplot()
p
```

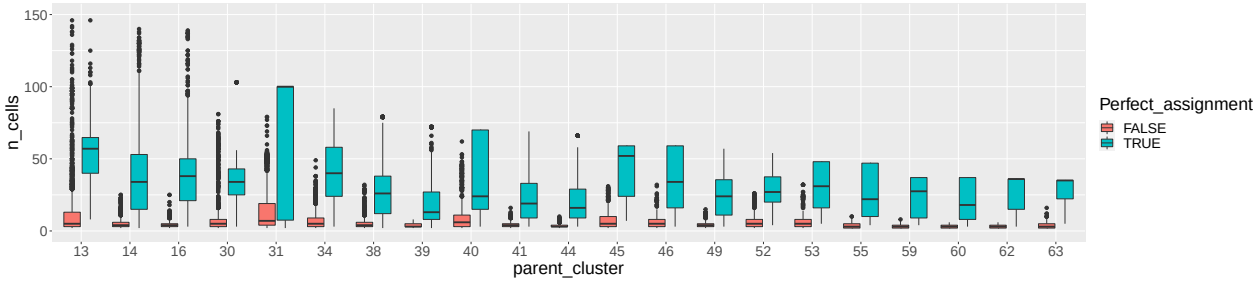
```
## Warning: Removed 88 rows containing non-finite values (stat_boxplot).
```



Number of cells for sub-clusters when the automatically assigned identities match exactly the expected

```
# Box plot
p <- ggplot(IterationData, aes(x = parent_cluster, y = n_cells,
    fill = Perfect_assignment)) + ylim(0, 150) + theme(text = element_text(size = 20)) +
    geom_boxplot()
p
```

```
## Warning: Removed 88 rows containing non-finite values (stat_boxplot).
```



Statistics for Clusters automaticaly assigned in the first round (Final data from clustering parameters PCs = 92 and Resol = 4)

Summary statistics and boxplot

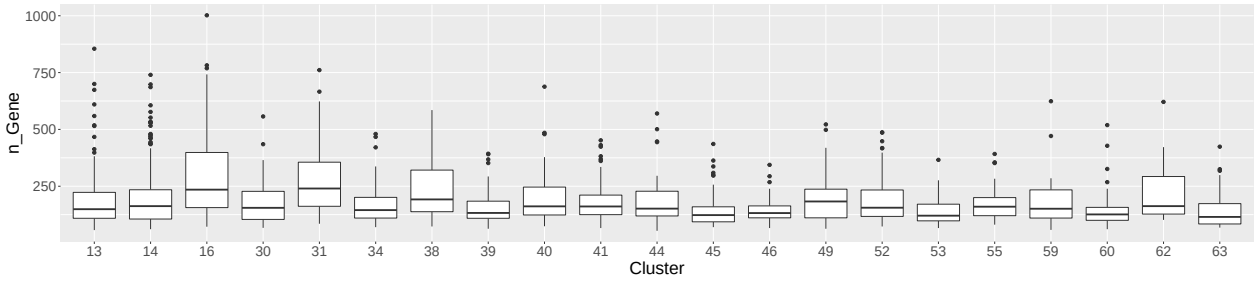
```
cells <- names(CeNeurons$seurat_clusters[CeNeurons$seurat_clusters %in%
    names(Clusters)])
dataClustering <- as.data.frame(CeNeurons$nFeature_RNA[cells])
dataClustering <- cbind(dataClustering, as.data.frame(CeNeurons$seurat_clusters[cells]))
colnames(dataClustering) <- c("n_Gene", "Cluster")

summaryData <- data.frame()
for (cluster in names(Clusters)) {
    summaryData <- rbind(summaryData, as.data.frame(as.list(summary(dataClustering[dataClustering$Cluste
        cluster, "n_Gene"]))))
}
summaryData <- cbind(Cluster = names(Clusters), summaryData)
summaryData
```

```
##    Cluster Min. X1st.Qu. Median     Mean X3rd.Qu. Max.
```

```
## 1       13   57   109.00  149.0 182.6904   223.00  855
## 2       14   61   105.75  162.5 199.3534   234.50  740
## 3       16   72   156.00  235.0 290.7854   398.50 1002
## 4       30   67   104.00  155.0 175.8252   227.50  557
## 5       31   85   161.75  240.0 279.7100   355.75  761
## 6       34   70   110.00  145.0 169.1059   201.00  480
## 7       38   73   138.00  192.0 234.2025   321.00  585
## 8       39   63   109.00  132.5 158.4028   184.25  393
## 9       40   74   123.50  161.5 197.6571   246.25  688
## 10      41   66   125.00  161.0 183.7681   211.00  452
## 11      44   54   119.25  151.5 178.9394   228.00  570
## 12      45   70    93.50  123.0 146.4915   159.50  436
## 13      46   66   111.00  132.0 143.7458   163.50  344
## 14      49   63   111.00  183.0 198.6842   237.00  522
## 15      52   73   117.25  155.5 196.7037   233.50  487
## 16      53   66    97.75  120.5 140.3750   171.00  366
## 17      55   81   120.50  160.0 174.5319   200.00  392
## 18      59   58   110.00  151.0 180.6486   234.00  624
## 19      60   61   100.00  126.0 154.1892   157.00  519
## 20      62  102   127.50  162.5 220.1389   292.75  621
## 21      63   68    84.00  115.0 146.7714   173.50  424
```

```
# Box plot
p <- ggplot(dataClustering, aes(x = Cluster, y = n_Gene)) + theme(text = element_text(size = 20)) +
    geom_boxplot()
p
```



Gene atlas summary

```
cured_brain_atlas_PL_and_OH_CircularOtherMarkers <- cured_brain_atlas_PL_and_OH_CircularOtherMarkers[,
    2:length(cured_brain_atlas_PL_and_OH_CircularOtherMarkers)]
nGene <- apply(cured_brain_atlas_PL_and_OH_CircularOtherMarkers[,
    unlist(Clusters)], 2, sum)
names(nGene) <- paste(unlist(names(Clusters)), names(nGene),
    sep = "_")
print("Total atlas genes per neuron class")
```

```
## [1] "Total atlas genes per neuron class"
```

```
nGene
```

```
##  13_CAN  14_AVK  16_FLP  30_RIA  31_ADL  34_ASG  38_ASJ  39_RMG  40_ASK  41_DVA
##      41      30      49      37      86      63      81      25      95      53
##  44_BAG  45_AIN  46_IL2  49_AWA  52_AVG  53_RIC  55_PVQ  59_RIS   60_M1 62_ASER
##      46      35      47      45      38      40      55      33      31      98
```

```
##   63_I5
##      32
```

```
specificGenes <- apply(cured_brain_atlas_PL_and_OH_CircularOtherMarkers,
    1, sum) == 1
cured_brain_atlas_specific <- cured_brain_atlas_PL_and_OH_CircularOtherMarkers[specificGenes,
    ]
nGene <- apply(cured_brain_atlas_specific[, unlist(Clusters)],
    2, sum)
names(nGene) <- paste(unlist(names(Clusters)), names(nGene),
    sep = "_")
print("Specific atlas genes (only expresed in one neuron)")
```

```
## [1] "Specific atlas genes (only expresed in one neuron)"
```

```
nGene
```

```
##  13_CAN  14_AVK  16_FLP  30_RIA  31_ADL  34_ASG  38_ASJ  39_RMG  40_ASK  41_DVA
##       1       1       3       2       4       1       3       1       5       2
##  44_BAG  45_AIN  46_IL2  49_AWA  52_AVG  53_RIC  55_PVQ  59_RIS   60_M1 62_ASER
##       3       0       2       3       0       1       3       2       0       2
##   63_I5
##       0
```