Pairwise-Distance-Unembedded-01-01-2021

Sixtus Dakurah

1/1/2021

```
# read in the data
pbmc_10X <- read.table("/Users/sixtusdakurah/Desktop/Liger/data/pbmc_10X.txt")
```

```
pbmc_SqW <- read.table("/Users/sixtusdakurah/Desktop/liger/data/pbmc_SeqWell.txt")
```

```
#head(pbmc_10X)
#head(pbmc_SqW)
dim(pbmc_10X)
```

```
## [1] 32738  2638
```

```
dim(pbmc_SqW)
```

```
## [1] 6713 3694
```

```
# get the cells from the two groups
pbmc_10X_Cells <- names(pbmc_10X)
print(length(unique(pbmc_10X_Cells)))
```

```
## [1] 2638
```

```
pbmc_SqW_Cells <- names(pbmc_SqW)
print(length(unique(pbmc_SqW_Cells)))
```

```
## [1] 3694
```

```
# load the cluster data
original_clusters <- read.csv("clusters.csv") #head(original_clusters)
#head(original_clusters)
dim(original_clusters)
```

```
## [1] 6332    2
```

```
# how many unique clusters do we have
print(length(unique(original_clusters$x))) # 14 unique clusters
```

```
## [1] 14
```

```r
# how many cells from 10x are in the cluster group
print(length(intersect(pbmc_10X_Cells, original_clusters$X)))
```

```
## [1] 2638
```

```r
# how many cells from sqw are in the cluster group
print(length(intersect(pbmc_SqW_Cells, original_clusters$X)))
```

```
## [1] 3694
```

What we can't do: compute the pairwise distances between cells across datasets in the same cluster group. This is because, for any given pair, they could have different level of gene expressions if they come from different data sets.

What we can do: compute the pairwise distances between cells in the same cluster group for each dataset.

Pairwise distances for cells in 10x

```r
# first scale the dataframe to columns to begin with
# scaling is by dividing the column values by the rmse for the particular column
pbmc_10X_scale <- data.frame(scale(pbmc_10X, center = FALSE)) # we need positive expressions
pbmc_SqW_scale <- data.frame(scale(pbmc_SqW, center = FALSE)) # we need positive expressions
```

```r
pairwiseClusterDistance <- function(data, clusters){
  # create an empty vector to store the average pairwise distances
  uniq.clusters <- unique(clusters$x)
  K <- length(uniq.clusters)
  average_pairwise <- c(rep(NA, length(uniq.clusters)))
  pair_wise_list  <- list()
  # for each unique cluster, compute the average pairwise distance
  for (k in 1:K){
    # select all cells that belongs to this cluster
    belongs <- clusters[clusters$x==uniq.clusters[k], ]
    # then select all cells from the expression matrix that are in belongs
    data_cells <- names(data)
    relevant_cells <- intersect(data_cells, belongs$X)
    print(length(relevant_cells))
    # now select columns based on relevant cells
    sub_data <- data[, relevant_cells]
    # conver to matrix and transpose for use in the distance function
    sub_matrix = t(as.matrix(sub_data))
    # add the avergae distance to the vector
    num_pairwsie = choose(length(relevant_cells), 2)
    pair_wise_dist  = dist(sub_matrix)
    pair_wise_list[[length(pair_wise_list)+1]]  <- list(as.vector(pair_wise_dist))
    average_pairwise[k]  = sum(pair_wise_dist)/num_pairwsie
  }
  return(list(average_pairwise, pair_wise_list))
}
```

```
pairwise_10x = pairwiseClusterDistance(pbmc_10X_scale, original_clusters)
```

```
## [1] 1151
## [1] 342
## [1] 479
## [1] 155
## [1] 308
## [1] 157
## [1] 32
## [1] 14
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
```

```
pairwise_SqW = pairwiseClusterDistance(pbmc_SqW_scale, original_clusters)
```

```
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 376
## [1] 634
## [1] 269
## [1] 104
## [1] 471
## [1] 1840
```

```
pairwise_df = data.frame("clusters" = unique(original_clusters$x))
pairwise_df$pbmc_10X = pairwise_10x[[1]]
pairwise_df$pbmc_SqW = pairwise_SqW[[1]]
pairwise_df
```

```
##             clusters pbmc_10X pbmc_SqW
## 1        CD4 T cells 112.7815      NaN
## 2            B cells 126.9410      NaN
## 3    CD14+ Monocytes 126.5652      NaN
## 4          NK cells 131.3212      NaN
## 5        CD8 T cells 125.0038      NaN
## 6  FCGR3A+ Monocytes 101.5362      NaN
## 7    Dendritic cells 108.5823      NaN
## 8     Megakaryocytes 150.6793      NaN
## 9              Bcell      NaN 53.80840
## 10               CD4      NaN 44.80650
## 11               CD8      NaN 41.81144
```

```
## 12                DC      NaN 58.85051
## 13                NK      NaN 40.07904
## 14           Myeloid      NaN 52.35705
```

```
# convert the list into a long df
uniq.clusters <- unique(original_clusters$x)
pair_wise_list_10X <- pairwise_10x[[2]]
pair_wise_list_SqW <- pairwise_SqW[[2]]

df_10XSQW = data.frame("cluster" = 'cluster', "distance" = 0, "dataset" = 'd')

K = length(uniq.clusters)

for(k in 1:K){
  ls1 = pair_wise_list_SqW[[k]][[1]]
  ls2 = pair_wise_list_10X[[k]][[1]]

  if (length(ls1)!=0){
    ds <- rep("SqW", length(ls1))
    cluster <- rep(uniq.clusters[k], length(ls1))
    df_sqw = data.frame("cluster" = cluster, "distance" = ls1, "dataset" = ds)

    df_10XSQW = rbind(df_10XSQW, df_sqw)
  }

  if (length(ls2)!=0){
    ds <- rep("10X", length(ls2))
    cluster <- rep(uniq.clusters[k], length(ls2))
    df_10x = data.frame("cluster" = cluster, "distance" = ls2, "dataset" = ds)
    df_10XSQW = rbind(df_10XSQW, df_10x)
  }
}
```

Repeat the same for variable genes

```
varG <- read.csv("data/varG.csv")
var_genes <- varG$x
pbmc_10X_scale_var <- pbmc_10X_scale[rownames(pbmc_10X_scale) %in% var_genes, ]
pbmc_SqW_scale_var <- pbmc_SqW_scale[rownames(pbmc_SqW_scale) %in% var_genes, ]
```

```
# write to excel
#write.csv(pbmc_10X_scale_var, file = "pbmc_10X_var.csv")
#write.csv(pbmc_SqW_scale_var, file = "pbmc_SqW_var.csv")
```

```
pairwise_10x_var = pairwiseClusterDistance(pbmc_10X_scale_var, original_clusters)
```

```
## [1] 1151
## [1] 342
## [1] 479
## [1] 155
## [1] 308
```

```
## [1] 157
## [1] 32
## [1] 14
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
```

```
pairwise_SqW_var = pairwiseClusterDistance(pbmc_SqW_scale_var, original_clusters)
```

```
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 376
## [1] 634
## [1] 269
## [1] 104
## [1] 471
## [1] 1840
```

```
pairwise_df_var = data.frame("clusters" = unique(original_clusters$x))
pairwise_df_var$pbmc_10X = pairwise_10x_var[[1]]
pairwise_df_var$pbmc_SqW = pairwise_SqW_var[[1]]
pairwise_df_var
```

```
##             clusters  pbmc_10X pbmc_SqW
## 1        CD4 T cells 106.90065      NaN
## 2            B cells 118.57928      NaN
## 3    CD14+ Monocytes 120.70389      NaN
## 4           NK cells 121.69970      NaN
## 5        CD8 T cells 118.26424      NaN
## 6  FCGR3A+ Monocytes  95.85122      NaN
## 7    Dendritic cells 102.80395      NaN
## 8     Megakaryocytes 139.55401      NaN
## 9              Bcell       NaN 44.74460
## 10               CD4       NaN 38.91065
## 11               CD8       NaN 36.16425
## 12                DC       NaN 51.81645
## 13                NK       NaN 34.59562
## 14          Myeloid       NaN 45.54125
```

```
uniq.clusters <- unique(original_clusters$x)
pair_wise_list_10X_var <- pairwise_10x_var[[2]]
pair_wise_list_SqW_var <- pairwise_SqW_var[[2]]
```

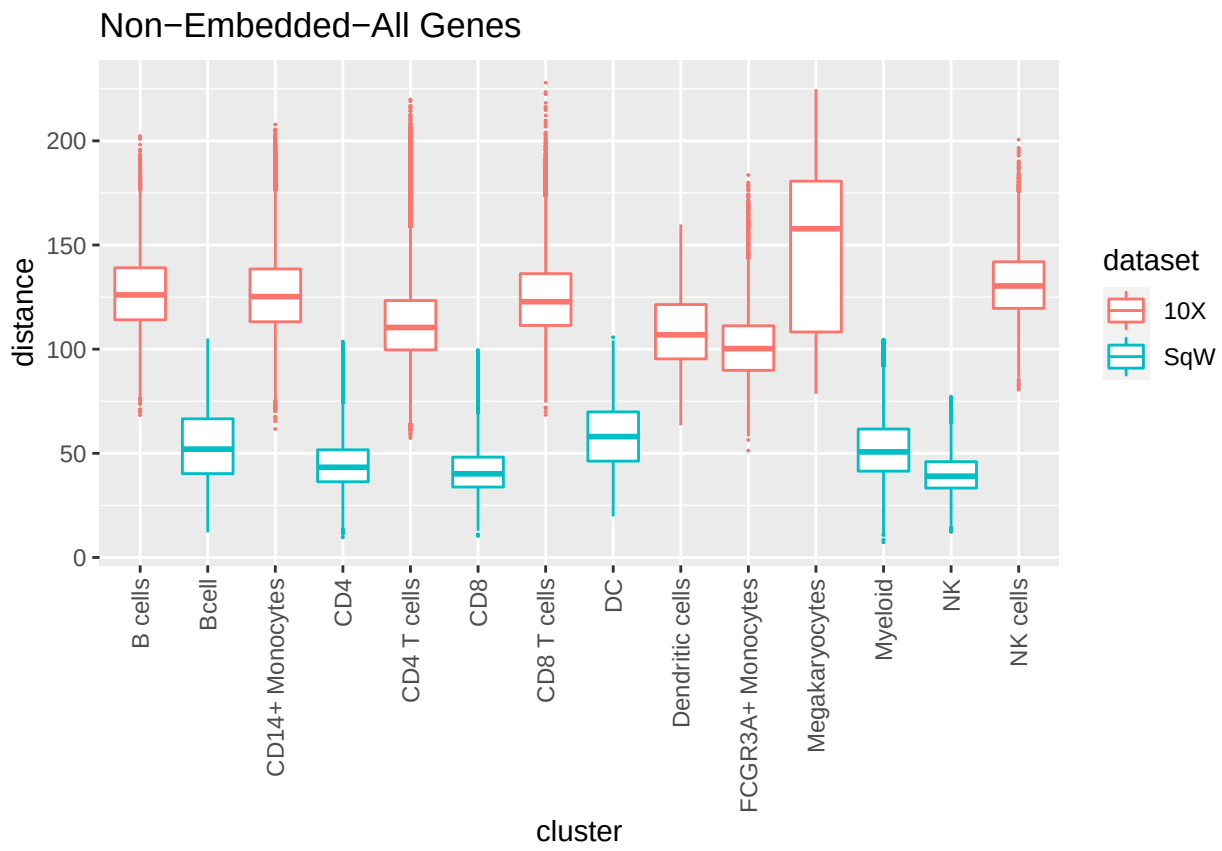
```
df_10XSQW_var = data.frame("cluster" = 'cluster', "distance" = 0, "dataset" = 'd')

K = length(uniq.clusters)

for(k in 1:K){
  ls1_var = pair_wise_list_SqW_var[[k]][[1]]
  ls2_var = pair_wise_list_10X_var[[k]][[1]]

  if (length(ls1_var)!=0){
    ds_var <- rep("SqW", length(ls1_var))
    cluster_var <- rep(uniq.clusters[k], length(ls1_var))
    df_sqw_var = data.frame("cluster" = cluster_var, "distance" = ls1_var, "dataset" = ds_var)

    df_10XSQW_var = rbind(df_10XSQW_var, df_sqw_var)
  }

  if (length(ls2_var)!=0){
    ds_var <- rep("10X", length(ls2_var))
    cluster_var <- rep(uniq.clusters[k], length(ls2_var))
    df_10x_var = data.frame("cluster" = cluster_var, "distance" = ls2_var, "dataset" = ds_var)
    df_10XSQW_var = rbind(df_10XSQW_var, df_10x_var)
  }
}
```

Now make the boxplots

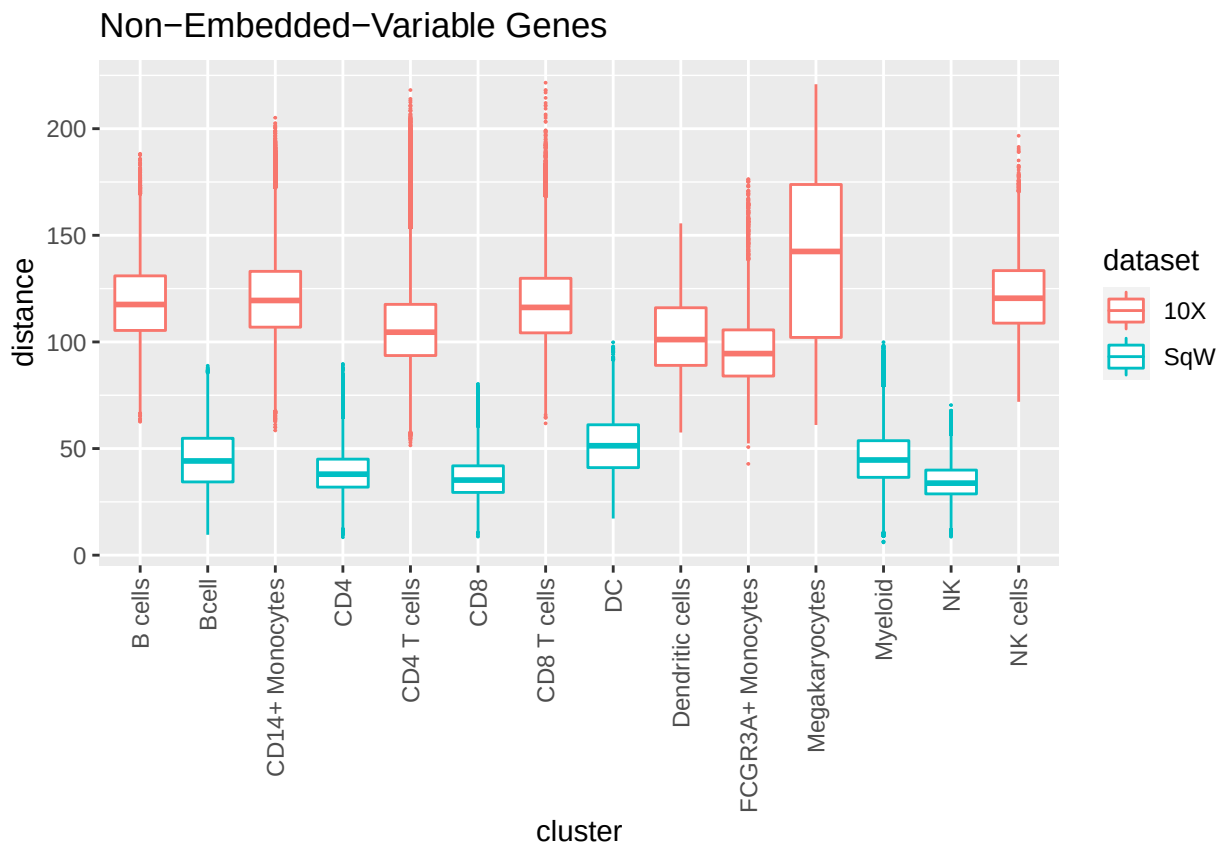
```
head(df_10XSQW)
```

```
##       cluster distance dataset
## 1     cluster   0.0000       d
## 2 CD4 T cells 125.8908     10X
## 3 CD4 T cells 137.2602     10X
## 4 CD4 T cells 112.2868     10X
## 5 CD4 T cells 107.9698     10X
## 6 CD4 T cells 121.1350     10X
```

```
library(ggplot2)
ggplot(data = df_10XSQW[-1, ]) +
  geom_boxplot(mapping = aes(x = cluster, y = distance, color = dataset), outlier.size = 0) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) + ggtitle("Non-Embedded-All Genes
```




```
ggplot(data = df_10XSQW_var[-1, ]) +
  geom_boxplot(mapping = aes(x = cluster, y = distance, color = dataset), outlier.size = 0) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) + ggtitle("Non-Embedded-Variable
```



Format the cell group names

```
## Monocyets (white blood cells) is exclusive to only 10x expressions
cell_groups <- list(
  c("B cells", "Bcell"),
  c("Megakaryocytes", "Myeloid"),
  c("CD4", "CD4 T cells"),
  c("CD8", "CD8 T cells"),
  c("DC", "Dentritic cells"),
  c("NK", "NK cells"))

original_clusters$group <- original_clusters$x
for(grp in cell_groups){
 original_clusters[original_clusters$x %in% grp, "group"] <- grp[1]
}
tail(original_clusters)
```

```
##                  X       x          group
## 6327 Myeloid_1835 Myeloid Megakaryocytes
## 6328 Myeloid_1836 Myeloid Megakaryocytes
## 6329 Myeloid_1837 Myeloid Megakaryocytes
## 6330 Myeloid_1838 Myeloid Megakaryocytes
## 6331 Myeloid_1839 Myeloid Megakaryocytes
## 6332 Myeloid_1840 Myeloid Megakaryocytes
```

```r
# Format the cells for SQwell expressions
extractCell <- function(x){
  substring(x, 0, regexpr("_", x) - 1)
}

pbmc_SqW_Cells_format <- lapply(pbmc_SqW_Cells, extractCell)
```

```r
# merge the dataset
#merged_data <- cbind(pbmc_10X_scale_var, pbmc_SqW_scale_var)
merged_data <- cbind(pbmc_10X_scale_var, pbmc_SqW_scale_var)

dim(merged_data)
```

```
## [1] 2814 6332
```

```r
#tail(merged_data[, 6329:6332])
```

```r
# add the labels to the cells for triplet loss training
labels_df <- rep("", dim(merged_data)[2])
names(labels_df) <- names(merged_data)
cellCount  = dim(merged_data)[2]

merged_labels <- rbind(merged_data, labels_df)
lastIndex = dim(merged_labels)[1]
#tail(merged_labels[, 6329:6332])
merged_names <- names(merged_labels)
```

```r
for (c in 1:cellCount){
  cell  = merged_names[c]
  # select the label from the clusters
  lab = (original_clusters[original_clusters$X==cell, ])$group
  if(length(lab) >= 1){
    merged_labels[lastIndex, cell] <- lab
  }

  #print(lab)
  # assign the lable
}
```

```r
rownames(merged_labels)[rownames(merged_labels) == "2815"] <- "Labels"
write.csv(merged_labels, file = "merged_with_labels.csv")
#tail(merged_labels[, 6331:6332])
```

```r
pairwiseClusterDistanceMerged <- function(data, clusters){
  # create an empty vector to store the average pairwise distances
  uniq.clusters <- unique(clusters$group)
  K <- length(uniq.clusters)
  average_pairwise <- c(rep(NA, length(uniq.clusters)))
  pair_wise_list  <- list()
  # for each unique cluster, compute the average pairwise distance
  for (k in 1:K){
```

```
    # select all cells that belongs to this cluster
    belongs <- clusters[clusters$group==uniq.clusters[k], ]
    # then select all cells from the expression matrix that are in belongs
    data_cells <- names(data)
    relevant_cells <- intersect(data_cells, belongs$X)
    # now select columns based on relevant cells
    sub_data <- data[, relevant_cells]
    # convert to matrix and transpose for use in the distance function
    sub_matrix = t(as.matrix(sub_data))
    # add the avergae distance to the vector
    num_pairwsie = choose(length(relevant_cells), 2)
    pair_wise_dist  = dist(sub_matrix)
    pair_wise_list[[length(pair_wise_list)+1]]  <- list(as.vector(pair_wise_dist))
    average_pairwise[k]  = sum(pair_wise_dist)/num_pairwsie
  }
  return(list(average_pairwise, pair_wise_list))
}
```

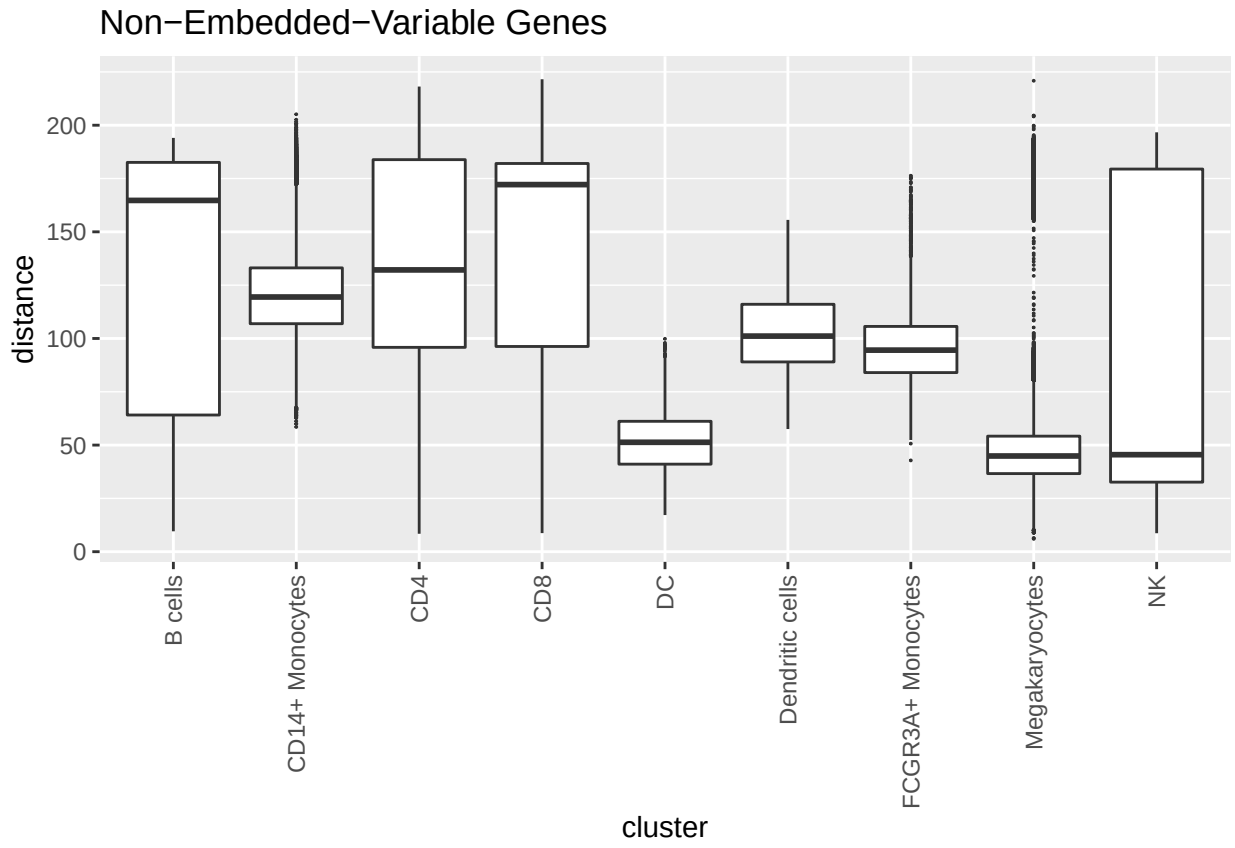
```
pairwise_10xSqW = pairwiseClusterDistanceMerged(merged_data, original_clusters)
```

```
pairwise_df_var_merged = data.frame("clusters" = unique(original_clusters$group))
pairwise_df_var_merged$pbmc_10Xpbmc_SqW = pairwise_10xSqW[[1]]
pairwise_df_var_merged
```

```
##            clusters pbmc_10Xpbmc_SqW
## 1              CD4        133.86264
## 2           B cells        130.43573
## 3   CD14+ Monocytes        120.70389
## 4                NK         94.23424
## 5              CD8        132.32877
## 6 FCGR3A+ Monocytes         95.85122
## 7   Dendritic cells        102.80395
## 8    Megakaryocytes         47.54651
## 9                DC         51.81645
```

```
uniq.clusters <- unique(original_clusters$group)
pair_wise_list_10XSqW_var <- pairwise_10xSqW[[2]]

df_10XSQW_var_merged = data.frame("cluster" = 'cluster', "distance" = 0)

K = length(uniq.clusters)

for(k in 1:K){
  ls_var = pair_wise_list_10XSqW_var[[k]][[1]]

  if (length(ls_var)!=0){
    cluster_var_10xsqwl <- rep(uniq.clusters[k], length(ls_var))
    df_10xsqw_var_merged = data.frame("cluster" = cluster_var_10xsqwl, "distance" = ls_var)

    df_10XSQW_var_merged = rbind(df_10XSQW_var_merged, df_10xsqw_var_merged)
  }
}
```

```
ggplot(data = df_10XSQW_var_merged[-1, ]) +
  geom_boxplot(mapping = aes(x = cluster, y = distance), outlier.size = 0) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) + ggtitle("Non-Embedded-Variable
```

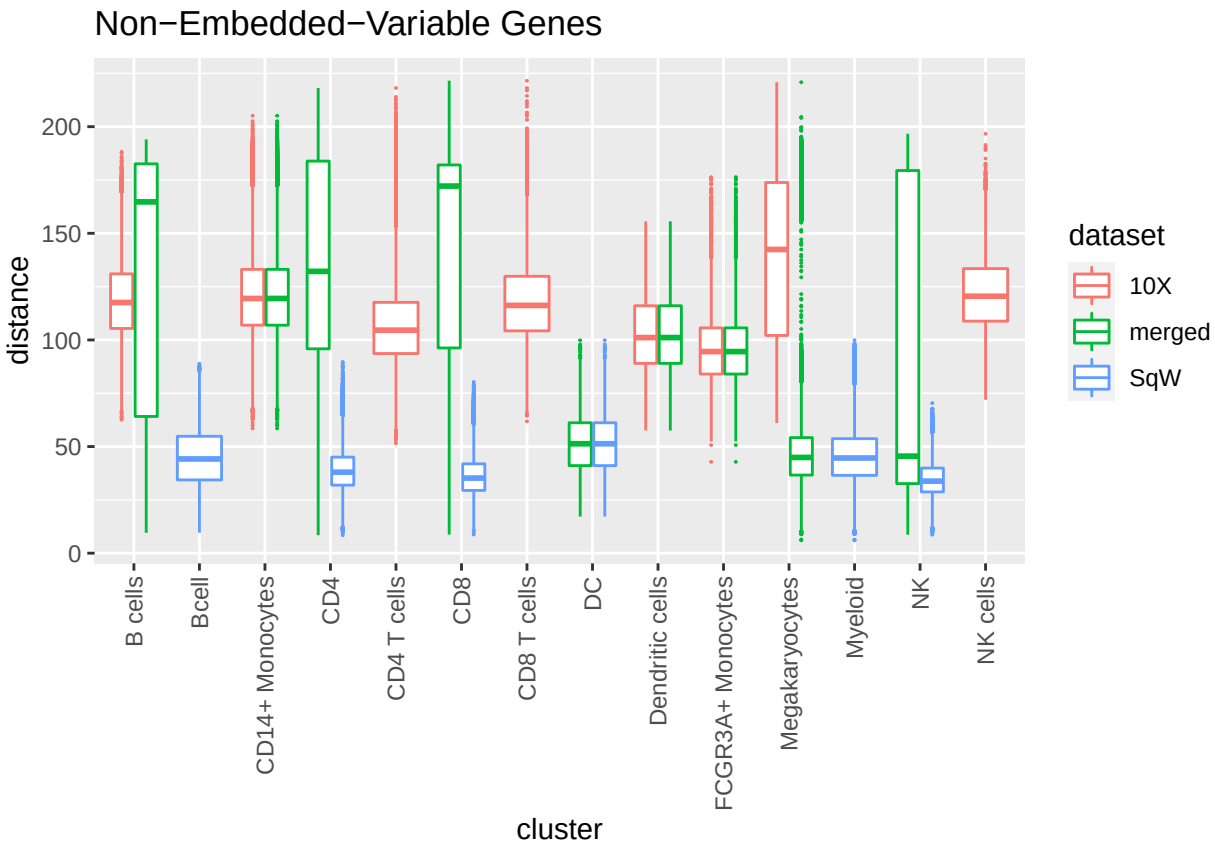


Now make a combined boxplot

```
df_10XSQW_var_merged$dataset <- rep("merged", dim(df_10XSQW_var_merged)[1])
all_pairwise <- rbind(df_10XSQW_var_merged, df_10XSQW_var[-1, ])
tail(all_pairwise)
```

```
##          cluster distance dataset
## 30217871 Myeloid 55.37748     SqW
## 30217881 Myeloid 55.39606     SqW
## 30217891 Myeloid 41.70333     SqW
## 30217901 Myeloid 45.84711     SqW
## 30217911 Myeloid 43.31870     SqW
## 30217921 Myeloid 50.01872     SqW
```

```
ggplot(data = all_pairwise[-1, ]) +
  geom_boxplot(mapping = aes(x = cluster, y = distance, color = dataset), outlier.size = 0) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) + ggtitle("Non-Embedded-Variable
```


Non−Embedded−Variable Genes
distance
200
150
100
50
0
B cells
Bcell
CD14+ Monocytes
CD4
CD4 T cells
CD8
CD8 T cells
DC
Dendritic cells
FCGR3A+ Monocytes
Megakaryocytes
Myeloid
NK
NK cells
cluster
dataset
10X
merged
SqW