Pairwise-Distance-Embedded-01-01-2021

Sixtus Dakurah

1/1/2021

```
pbmc_10X_c <- read.table("/Users/sixtusdakurah/Desktop/Liger/data/pbmc_10X.txt")
pbmc_SqW_c <- read.table("/Users/sixtusdakurah/Desktop/liger/data/pbmc_SeqWell.txt")
pbmc_10X_Cells_c <- names(pbmc_10X_c)
pbmc_SqW_Cells_c <- names(pbmc_SqW_c)
```

```
library(data.table)
# read in the data
pbmc_10X <- transpose((read.csv("/Users/sixtusdakurah/Desktop/Liger/data/H1-01-02-2021.csv"))[, -1])
pbmc_10X <- pbmc_10X[, -dim(pbmc_10X)[2]]
colnames(pbmc_10X) <- pbmc_10X_Cells_c
```

```
pbmc_SqW <- transpose((read.csv("/Users/sixtusdakurah/Desktop/liger/data/H2-01-02-2021.csv"))[, -1])
pbmc_SqW <- pbmc_SqW[, -dim(pbmc_SqW)[2]]
colnames(pbmc_SqW) <- pbmc_SqW_Cells_c
```

```
pbmc_10X[pbmc_10X < 0] <- 0
pbmc_SqW[pbmc_SqW < 0] <- 0
#(tail(pbmc_10X))[, 2634:2638]
#head(pbmc_SqW)
dim(pbmc_10X)
```

```
## [1]   20 2638
```

```
dim(pbmc_SqW)
```

```
## [1]   20 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
```

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

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

```r
# 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 <- pbmc_10X#data.frame(scale(pbmc_10X, center = FALSE)) # we need positive expressions
pbmc_SqW_scale <- pbmc_SqW#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)
pairwise_SqW = pairwiseClusterDistance(pbmc_SqW_scale, original_clusters)
```

```
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 0.2114869       NaN
## 2            B cells 0.2120294       NaN
## 3    CD14+ Monocytes 0.2110064       NaN
## 4           NK cells 0.1984400       NaN
## 5        CD8 T cells 0.2144462       NaN
## 6  FCGR3A+ Monocytes 0.2223662       NaN
## 7    Dendritic cells 0.2055878       NaN
## 8     Megakaryocytes 0.2011860       NaN
## 9              Bcell       NaN 0.1139082
## 10               CD4       NaN 0.1139804
## 11               CD8       NaN 0.1130044
## 12                DC       NaN 0.1108827
## 13                NK       NaN 0.1020036
## 14          Myeloid       NaN 0.1082880
```

```
# 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 – will not change anything

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

```
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 0.2114869       NaN
## 2            B cells 0.2120294       NaN
## 3    CD14+ Monocytes 0.2110064       NaN
## 4           NK cells 0.1984400       NaN
## 5        CD8 T cells 0.2144462       NaN
## 6  FCGR3A+ Monocytes 0.2223662       NaN
## 7    Dendritic cells 0.2055878       NaN
## 8     Megakaryocytes 0.2011860       NaN
## 9              Bcell       NaN 0.1139082
## 10               CD4       NaN 0.1139804
## 11               CD8       NaN 0.1130044
## 12                DC       NaN 0.1108827
## 13                NK       NaN 0.1020036
## 14          Myeloid       NaN 0.1082880
```

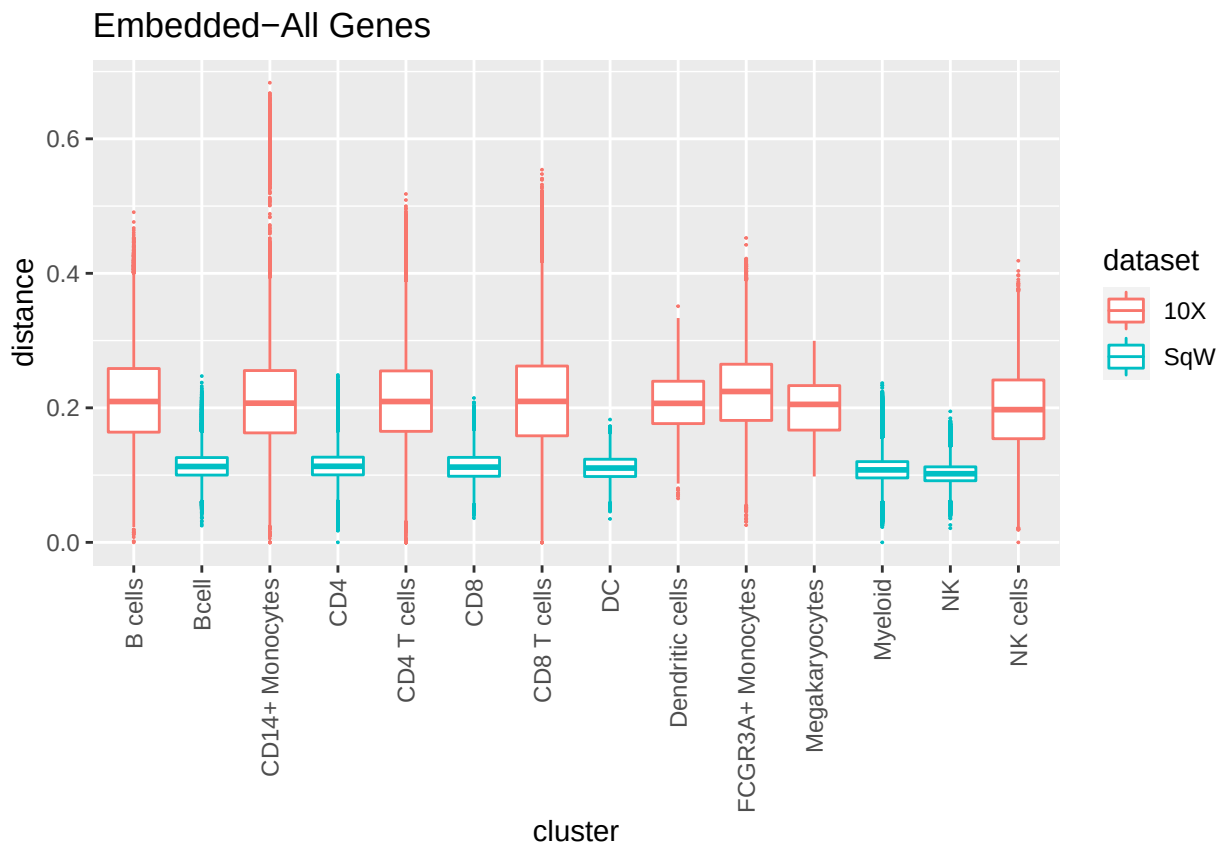
```
uniq.clusters <- unique(original_clusters$x)
pair_wise_list_10X_var <- pairwise_10x_var[[2]]
pair_wise_list_SqW_var <- pairwise_SqW_var[[2]]

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

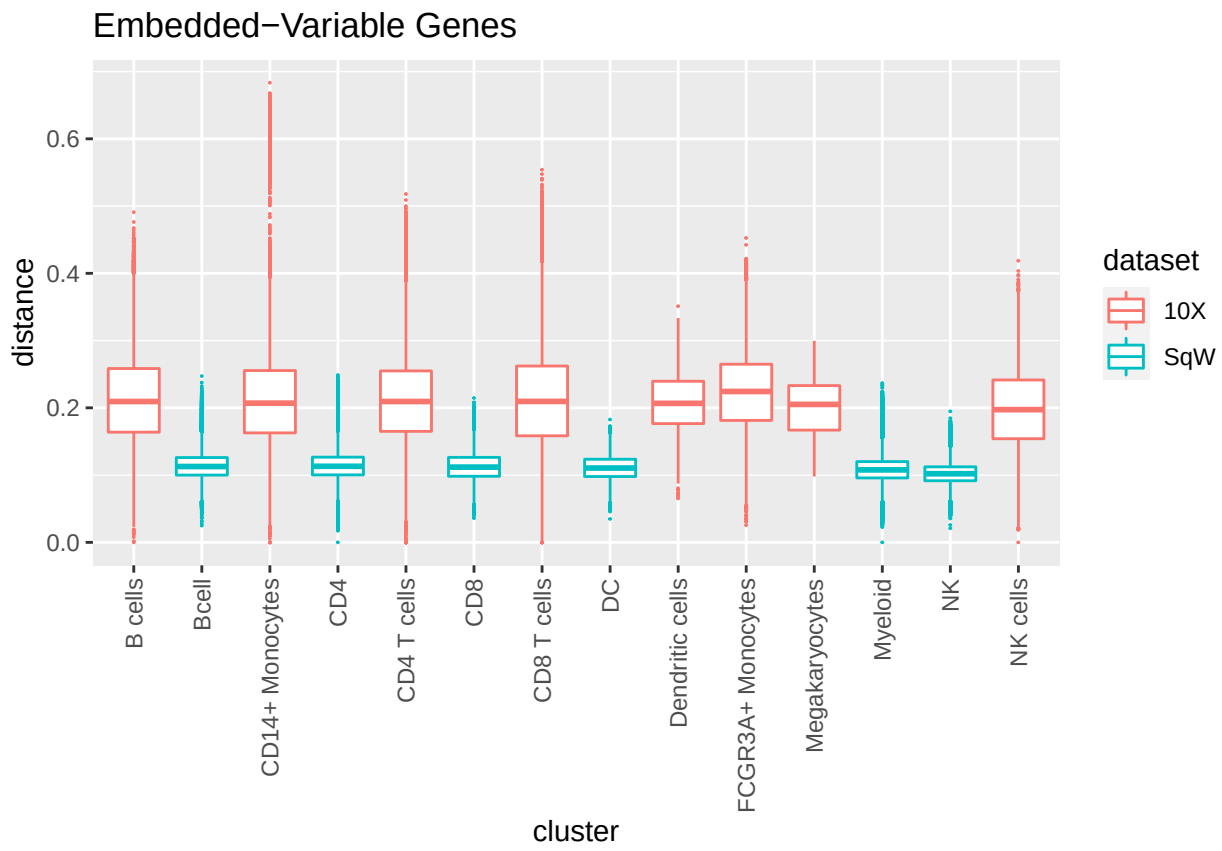
```
head(df_10XSQW)
```

```
##       cluster  distance dataset
## 1     cluster 0.0000000       d
## 2 CD4 T cells 0.2944659     10X
## 3 CD4 T cells 0.1659634     10X
## 4 CD4 T cells 0.1747941     10X
## 5 CD4 T cells 0.2495497     10X
## 6 CD4 T cells 0.2363102     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("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("Embedded-Variable Genes
```



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
```

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

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

```
# 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]   20 6332
```

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

```
# 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])
```

```
##    Myeloid_1837 Myeloid_1838 Myeloid_1839 Myeloid_1840
## 16  0.012465225  0.010767697            0   0.03111582
## 17  0.015167502            0 0.0041316403  0.039905764
## 18            0            0            0            0
## 19  0.032999482            0  0.032693055   0.04391579
## 20   0.04101604            0  0.037744798            0
## 21
```

```
merged_names <- names(merged_labels)
```

```
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){
    if(is.na(cell)){
      print(cell)
      print(c)
    }else{
    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))
}
```

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

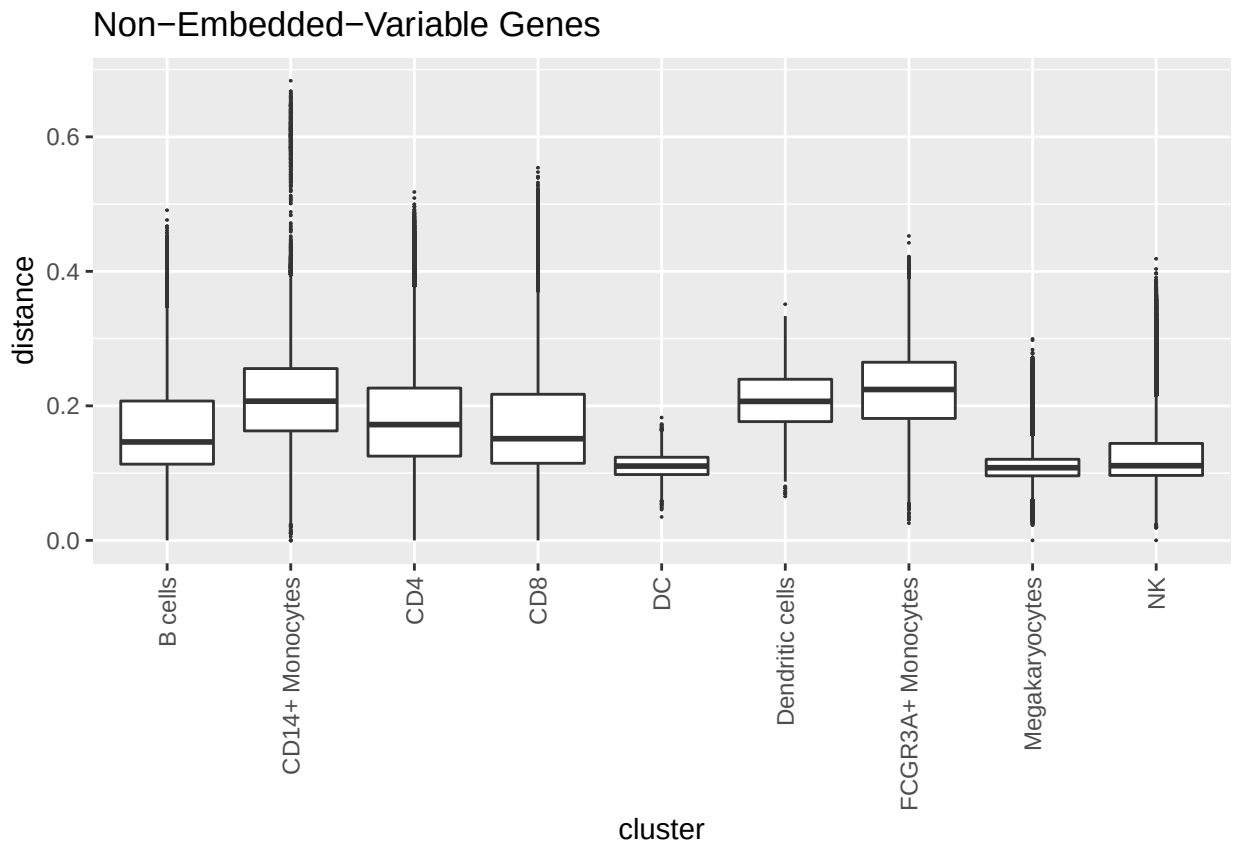
```r
pairwise_df_var_merged = data.frame("clusters" = unique(original_clusters$group))
pairwise_df_var_merged$pbmc_10Xpbmc_SqW = pairwise_10xSqW[[1]]
pairwise_df_var_merged
```

```
##            clusters pbmc_10Xpbmc_SqW
## 1               CD4        0.1810092
## 2           B cells        0.1651716
## 3   CD14+ Monocytes        0.2110064
## 4                NK        0.1291940
## 5               CD8        0.1714580
## 6 FCGR3A+ Monocytes        0.2223662
## 7   Dendritic cells        0.2055878
## 8    Megakaryocytes        0.1091297
## 9                DC        0.1108827
```

```r
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 0.12360884     SqW
## 30217881 Myeloid 0.06726201     SqW
## 30217891 Myeloid 0.09351275     SqW
## 30217901 Myeloid 0.11923383     SqW
## 30217911 Myeloid 0.11835702     SqW
## 30217921 Myeloid 0.09835778     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("Embedded-Variable Genes
```