Task1

Task 1. (5%) Prepare all scRNA-seq count data into a single R object.

Load all samples and create a SingleCellExperiment object that contains necessary column data to annotate the experimental design. Report the exact numbers of cells in each sample.

```
# Combine the count matrices and metadata into lists

counts_list <- list(P15_counts, P30_counts, P40_counts, P50_counts, P70_counts, Adult_counts, Adu
metadata_list <- list(P15_meta, P30_meta, P40_meta, P50_meta, P70_meta, Adult_meta, Adult_male_me



# lapply(counts_list[1:1], str)
lapply(metadata_list, head, n = 3)
```

```
[[1]]
                  orig.ident nCount_RNA nFeature_RNA nCount_integrated
1_AAACCTGAGGACGAAA          1       5341         1873                NA
1_AAACCTGAGGCACATG          1       4725         1676                NA
1_AAACCTGAGGTAGCCA          1       8118         2229                NA
                  nFeature_integrated integrated_snn_res.6 Clustering NNPreds
1_AAACCTGAGGACGAAA                  NA                   23          2       7
1_AAACCTGAGGCACATG                  NA                   44        101     123
1_AAACCTGAGGTAGCCA                  NA                  104         31      30
                  NNPredsConfidence FinalIdents
1_AAACCTGAGGACGAAA             0.714           7
1_AAACCTGAGGCACATG             0.478         123
1_AAACCTGAGGTAGCCA             0.982          30

[[2]]
                  orig.ident nCount_RNA nFeature_RNA nCount_integrated
1_AAACCTGAGCTAACTC          1       4663         1766                NA
1_AAAGATGTCACTATTC          1       2328         1163                NA
1_AACACGTGTCAGAAGC          1       4421         1653                NA
                  nFeature_integrated integrated_snn_res.6 Clustering NNPreds
1_AAACCTGAGCTAACTC                  NA                   13         54      61
1_AAAGATGTCACTATTC                  NA                   13         54      61
1_AACACGTGTCAGAAGC                  NA                   13         54      61
                  NNPredsConfidence FinalIdents
1_AAACCTGAGCTAACTC             0.892          61
1_AAAGATGTCACTATTC             0.950          61
1_AACACGTGTCAGAAGC             0.990          61

[[3]]
```

```
                   orig.ident nCount_RNA nFeature_RNA nCount_integrated
1_AAACCTGAGACTTTCG          1      10915         2452                NA
1_AACTCTTTCCATGAAC          1       1307          623                NA
1_ACGAGCCGTAAATACG          1       4964         1494                NA
                   nFeature_integrated integrated_snn_res.6 Clustering NNPreds
1_AAACCTGAGACTTTCG                  NA                   94        121     206
1_AACTCTTTCCATGAAC                  NA                   94        121     206
1_ACGAGCCGTAAATACG                  NA                   94        121     206
                   NNPredsConfidence FinalIdents
1_AAACCTGAGACTTTCG             0.982         206
1_AACTCTTTCCATGAAC             0.880         206
1_ACGAGCCGTAAATACG             0.946         206

[[4]]
                   orig.ident nCount_RNA nFeature_RNA nCount_integrated
1_AAACCTGAGCCACGCT          1       2674          991                NA
1_AACACGTGTATTCTCT          1       1083          518                NA
1_AACCGCGGTTCTCATT          1      12407         2939                NA
                   nFeature_integrated integrated_snn_res.6 Clustering NNPreds
1_AAACCTGAGCCACGCT                  NA                   37          7     204
1_AACACGTGTATTCTCT                  NA                   37          7     200
1_AACCGCGGTTCTCATT                  NA                   58         80      66
                   NNPredsConfidence FinalIdents FinalIdents.T45separate
1_AAACCTGAGCCACGCT             0.824         204                     204
1_AACACGTGTATTCTCT             0.510           0                       0
1_AACCGCGGTTCTCATT             0.730          66                      66

[[5]]
                   orig.ident nCount_RNA nFeature_RNA nCount_integrated
5_AAACCTGAGAGACGAA          5       8947         2122                NA
5_AAACGGGCATACGCCG          5      10388         2391                NA
5_AAAGTAGTCCTTGACC          5      17681         3017                NA
                   nFeature_integrated integrated_snn_res.6 Clustering NNPreds
5_AAACCTGAGAGACGAA                  NA                   33         69      24
5_AAACGGGCATACGCCG                  NA                   33         69      24
5_AAAGTAGTCCTTGACC                  NA                   33         69     114
                   NNPredsConfidence FinalIdents
5_AAACCTGAGAGACGAA             0.902          24
5_AAACGGGCATACGCCG             0.736          24
5_AAAGTAGTCCTTGACC             1.000         114

[[6]]
                     orig.ident nCount_RNA nFeature_RNA percent.mito male_cells
AAACCTGAGTTAACGA_1 Adult_1d_N_1       1042          557   0.06333973          0
AAACCTGCATCGGACC_1 Adult_1d_N_1       2356         1112   0.02504244          0
AAACCTGGTACCGAGA_1 Adult_1d_N_1       2056          932   0.01945525          0
                   nCount_integrated nFeature_integrated integrated_snn_res.10
AAACCTGAGTTAACGA_1                NA                  NA                    1
AAACCTGCATCGGACC_1                NA                  NA                   17
AAACCTGGTACCGAGA_1                NA                  NA                   76
                   original.IDs Clustering FinalIdents
```

```
AAACCTGAGTTAACGA_1            1        133         234
AAACCTGCATCGGACC_1           17        125         125
AAACCTGGTACCGAGA_1           76        122         227

[[7]]
                     orig.ident nCount_RNA nFeature_RNA percent.mito male_cells
AAACCTGAGAAACGAG_1 Adult_1d_N_1       3502         1391   0.05282696          1
AAACCTGAGAAGAAGC_1 Adult_1d_N_1       1158          620   0.09758204          1
AAACCTGAGCCAACAG_1 Adult_1d_N_1       5305         1705   0.04033930          1
```

```r
remove(P15_counts, P30_counts, P40_counts, P50_counts, P70_counts, Adult_counts, Adult_male_counts
remove(P15_meta, P30_meta, P40_meta, P50_meta, P70_meta, Adult_meta, Adult_male_meta)

# Perform garbage collection
gc()
```

```
           used   (Mb) gc trigger    (Mb)   max used    (Mb)
Ncells   3587089  191.6    6719584   358.9    5147664   275.0
Vcells 545911222 4165.0 2181599878 16644.3 2956409541 22555.7
```

```r
  # Aligning genes across all samples
  all_genes <- Reduce(intersect, lapply(counts_list, rownames))
  counts_list_aligned <- lapply(counts_list, function(cts) {
    cts[all_genes, , drop = FALSE]
  })

# Aligning columns of metadata
common_meta_cols <- Reduce(intersect, lapply(metadata_list, colnames))
metadata_list_aligned <- lapply(metadata_list, function(df) {
  df[, common_meta_cols]
})

# Combining counts and metadata
all_counts_combined <- do.call(cbind, counts_list_aligned)
all_metadata_combined <- do.call(rbind, metadata_list_aligned)

# Creating a SingleCellExperiment object
sce <- SingleCellExperiment(assays = list(counts = all_counts_combined))
rowData(sce) <- DataFrame(all_genes)
colData(sce) <- DataFrame(all_metadata_combined)

# Ensuring that row names and column names in rowData and colData match counts and metadata
rownames(sce) <- all_genes
colnames(sce) <- rownames(all_metadata_combined)

# Reporting the number of cells in each sample
sample_cell_counts <- lapply(counts_list, ncol)
```

```
sample_cell_counts
```

```
[[1]]
[1] 31018

[[2]]
[1] 35758

[[3]]
[1] 24084

[[4]]
[1] 31340

[[5]]
[1] 43740

[[6]]
[1] 109743

[[7]]
[1] 11133
```

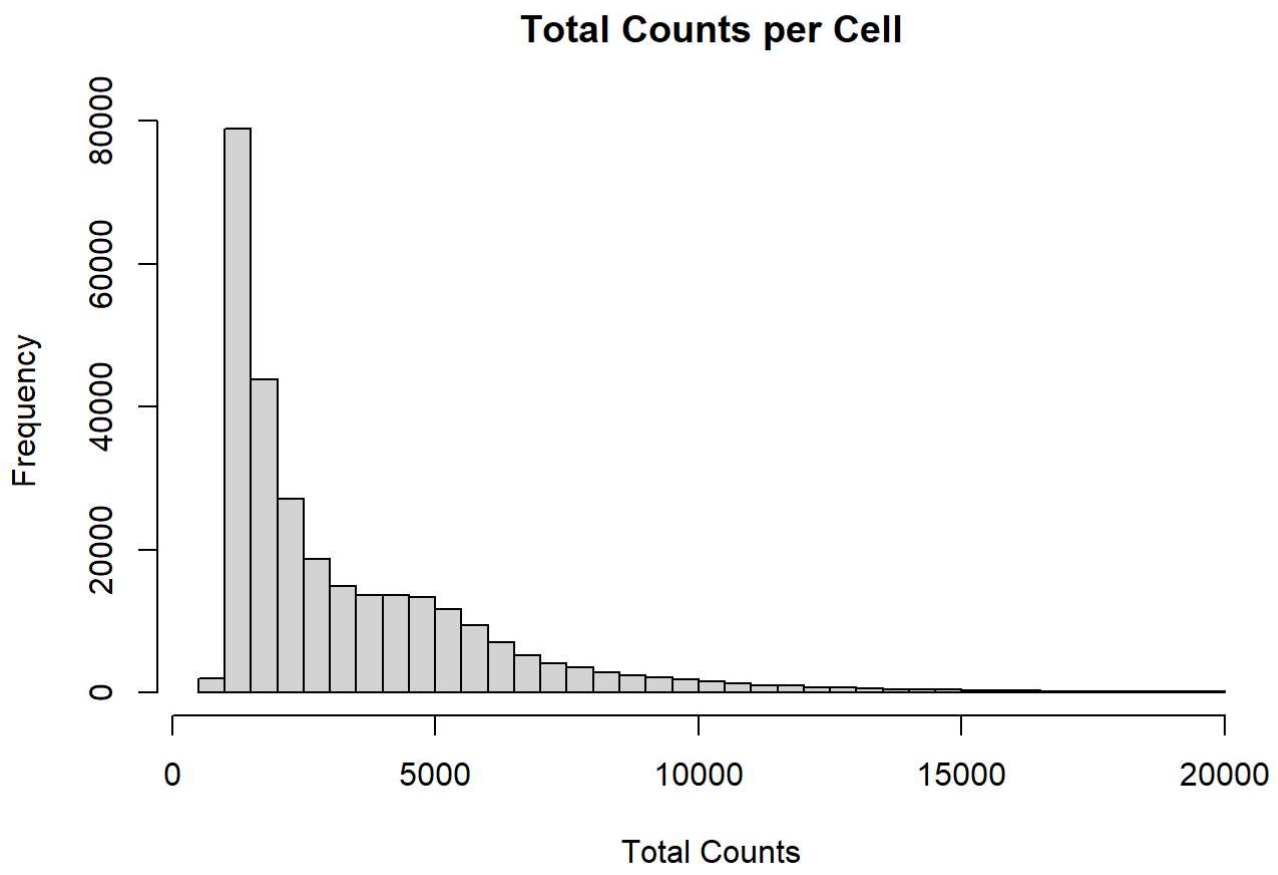
```
remove(all_counts_combined,counts_list_aligned,counts_list)
remove(all_metadata_combined)

# Perform garbage collection
gc()
```

```
           used   (Mb) gc trigger    (Mb)   max used    (Mb)
Ncells   8294877  443.0   15243398   814.1   11706240   625.2
Vcells 553320756 4221.6 2010703248 15340.5 2956409541 22555.7
```

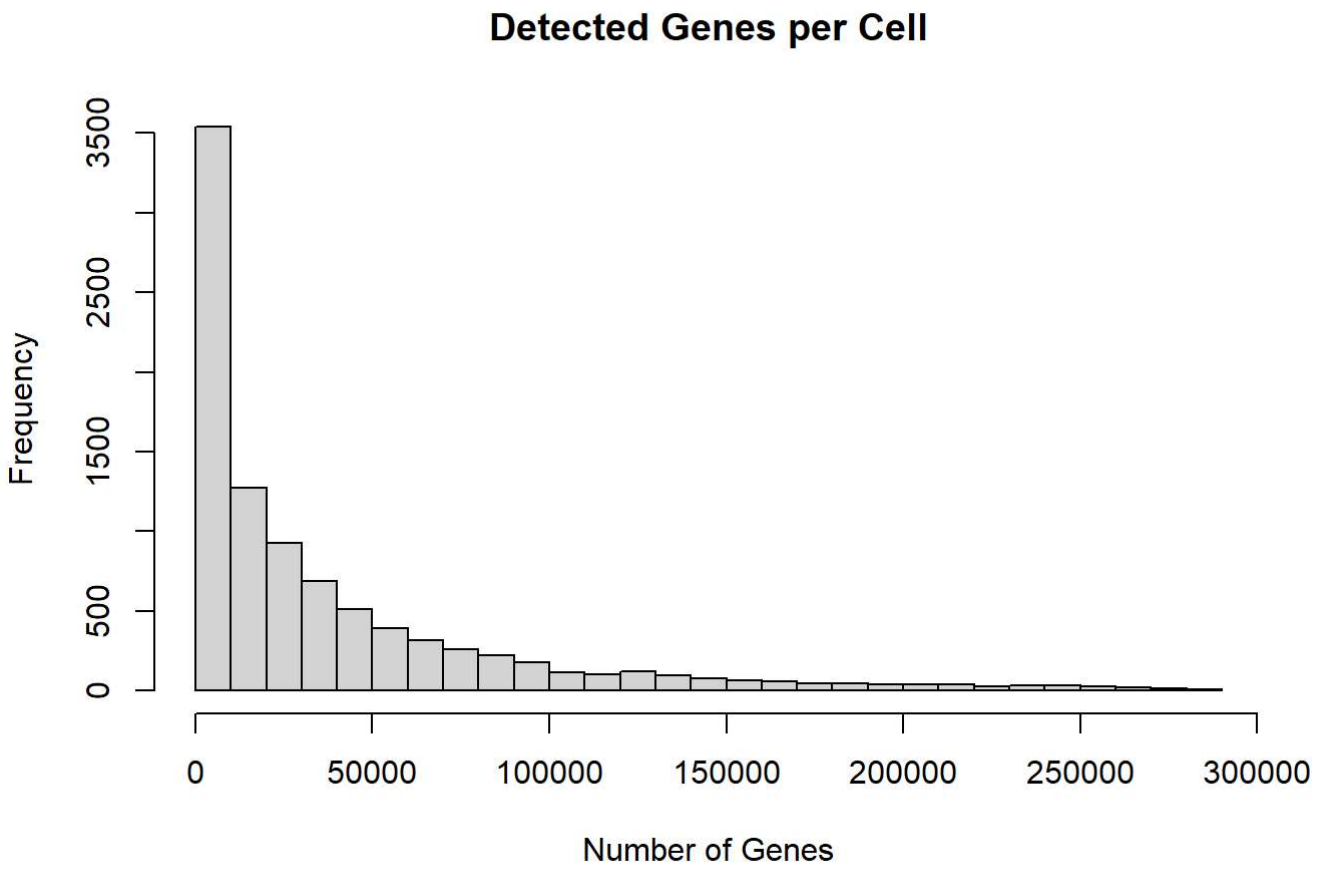
Task 2. (35%) Perform cell quality control

2.1 Show histograms of total counts of cells and number of detected genes in cells.

```
# Calculate total counts per cell
total_counts_per_cell <- colSums(counts(sce))

# Calculate the number of detected genes per cell
detected_genes_per_cell <- rowSums(counts(sce) > 0)

# Plot histograms
hist(total_counts_per_cell, main = "Total Counts per Cell", xlab = "Total Counts", breaks = 30)
```




```
hist(detected_genes_per_cell, main = "Detected Genes per Cell", xlab = "Number of Genes", breaks =
```



2.2 Remove cells of low total counts or number of detected genes. Report how many cells are removed.

```
cat("Number of cells in sce:", ncol(sce), "\n")
```

```
Number of cells in sce: 286816
```

```
cat("Number of genes in sce:", nrow(sce), "\n")
```

```
Number of genes in sce: 9293
```

```
# Calculate total counts per cell
total_counts_per_cell <- colSums(counts(sce))

# Calculate the number of detected genes per cell
detected_genes_per_cell <- rowSums(counts(sce) > 0)

# Check lengths
cat("Length of total_counts_per_cell:", length(total_counts_per_cell), "\n")
```

```
Length of total_counts_per_cell: 286816
```

```
cat("Length of detected_genes_per_cell:", length(detected_genes_per_cell), "\n")
```

```
Length of detected_genes_per_cell: 9293
```

```
# Correct calculation for the number of detected genes per cell
detected_genes_per_cell <- colSums(counts(sce) > 0)

# Check lengths again
cat("Length of total_counts_per_cell:", length(total_counts_per_cell), "\n")
```

```
Length of total_counts_per_cell: 286816
```

```
cat("Length of detected_genes_per_cell:", length(detected_genes_per_cell), "\n")
```

```
Length of detected_genes_per_cell: 286816
```

```
# Calculate total counts per cell
total_counts_per_cell <- colSums(counts(sce))

# Calculate the number of detected genes per cell
detected_genes_per_cell <- rowSums(counts(sce) > 0)

# Define thresholds
# You can adjust these thresholds based on your dataset and analysis needs
threshold_counts <- median(total_counts_per_cell) / 2
threshold_genes <- median(detected_genes_per_cell) / 2

# Filter cells
filtered_sce <- sce[, total_counts_per_cell > threshold_counts & detected_genes_per_cell > thresh
```

```
Warning in total_counts_per_cell > threshold_counts & detected_genes_per_cell >
: 长的对象长度不是短的对象长度的整倍数
```

```
# Report the number of cells removed
cells_removed <- ncol(sce) - ncol(filtered_sce)
cat("Number of cells removed:", cells_removed, "\n")
```

```
Number of cells removed: 126720
```

2.3 Remove genes that are zero in all samples. Report how many genes are removed.

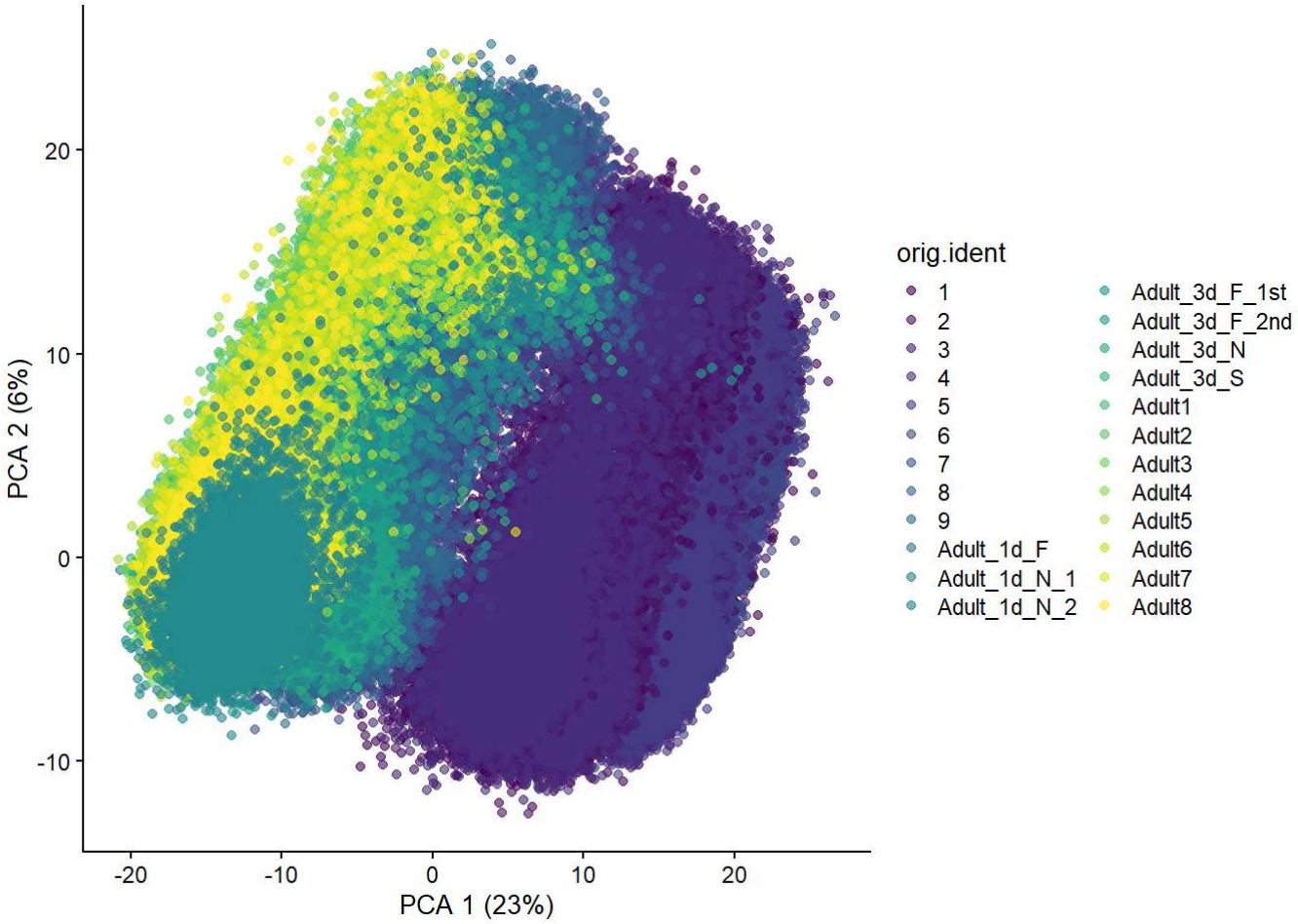
```
# Find genes that are not expressed in any cells
non_expressed_genes <- rowSums(counts(sce) > 0) == 0

# Remove these genes
```

```
filtered_sce <- filtered_sce[!non_expressed_genes, ]

# Report the number of genes removed
cat("Removed genes:", sum(non_expressed_genes), "\n")
```

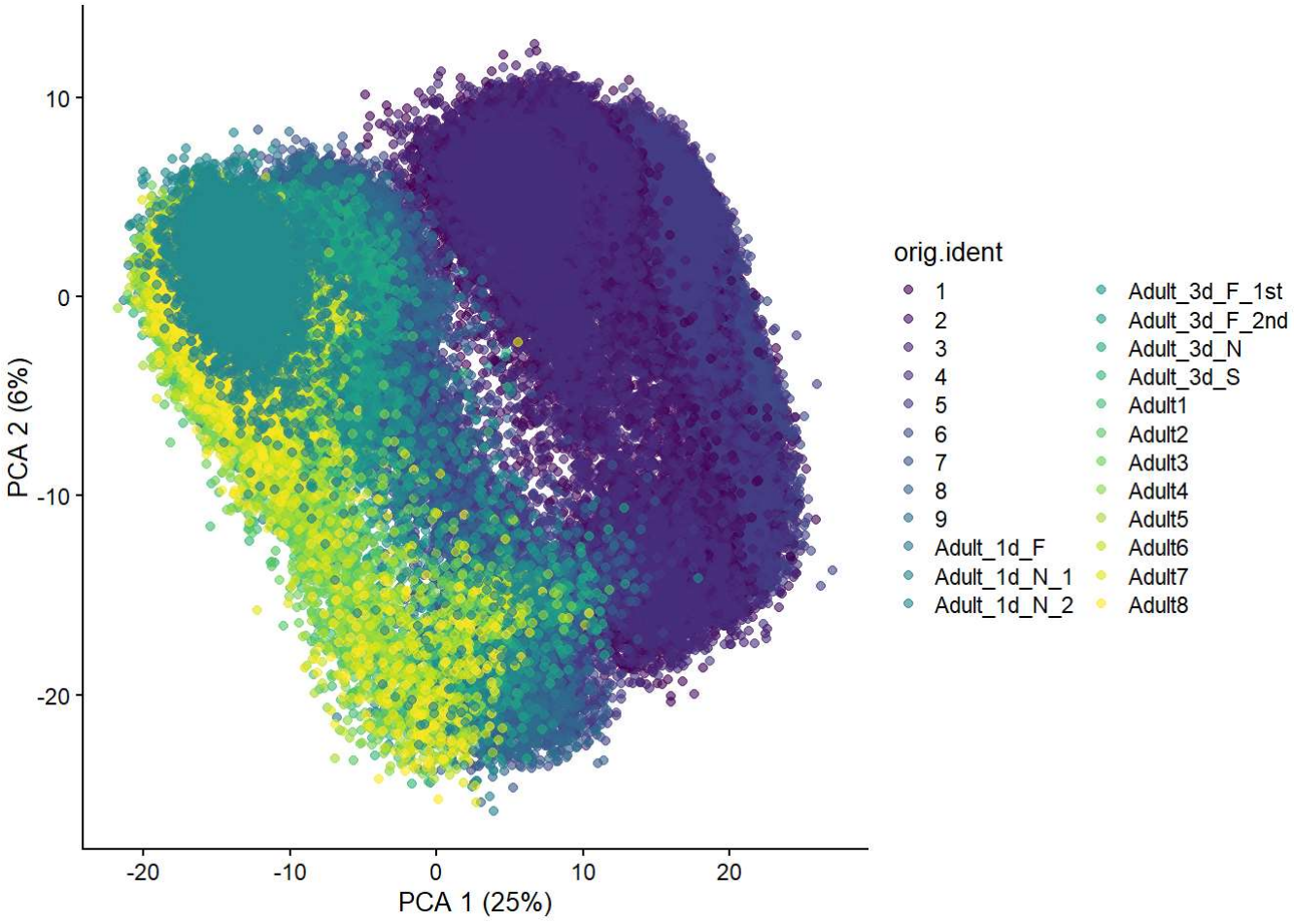
```
Removed genes: 0
```

2.4 Show PCA before and after quality control.

```
# Log-normalize the counts if not already done
sce <- logNormCounts(sce)
filtered_sce <- logNormCounts(filtered_sce)

# Run PCA on both datasets
sce <- runPCA(sce, ncomponents = 2)
filtered_sce <- runPCA(filtered_sce, ncomponents = 2)

# Plot PCA for original  data
plotPCA(sce, colour_by = "orig.ident")
```




```
# Plot PCA for quality-controlled data
plotPCA(filtered_sce, colour_by = "orig.ident")
```

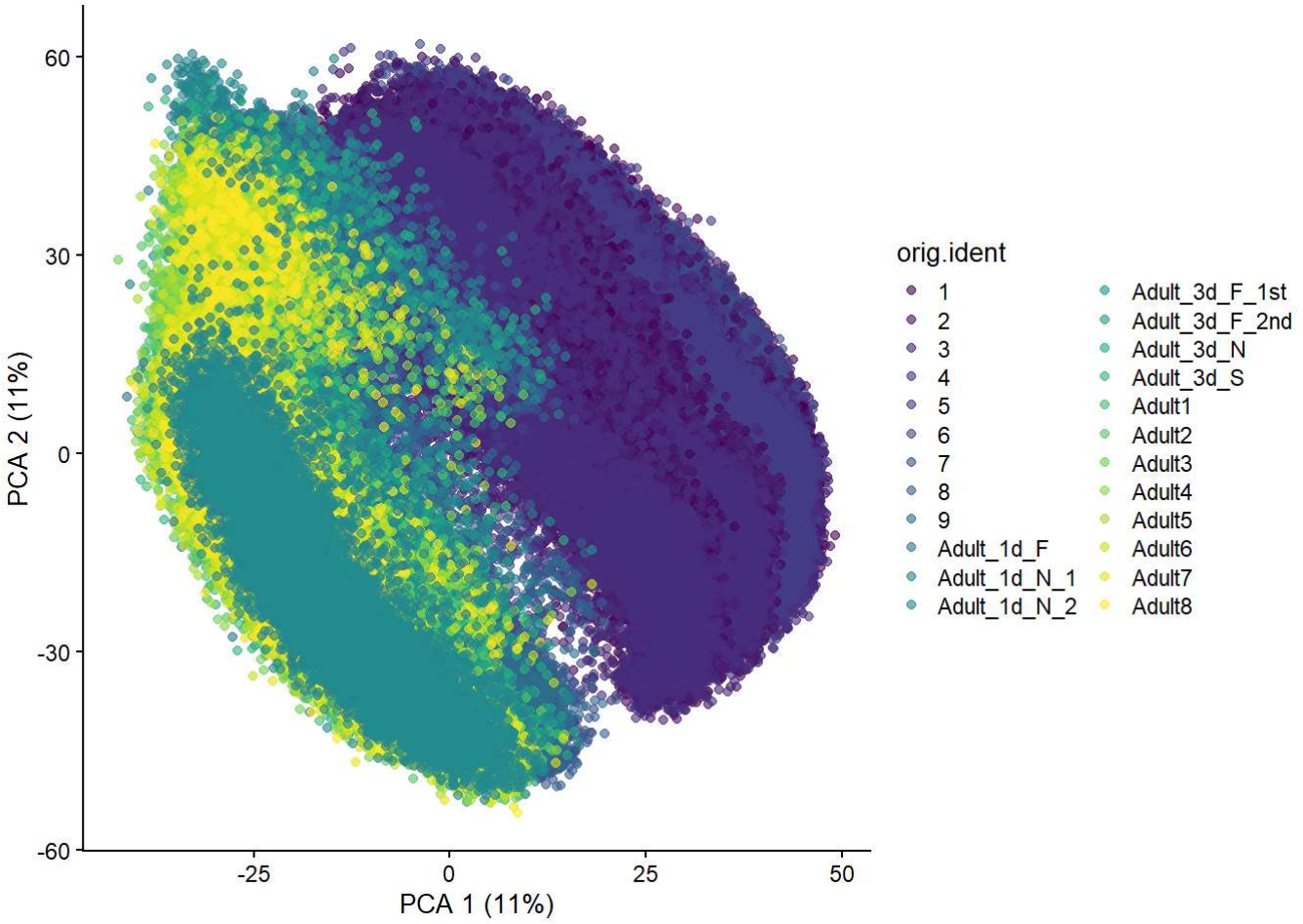


Task 3. (35%) Carry out library size normalization

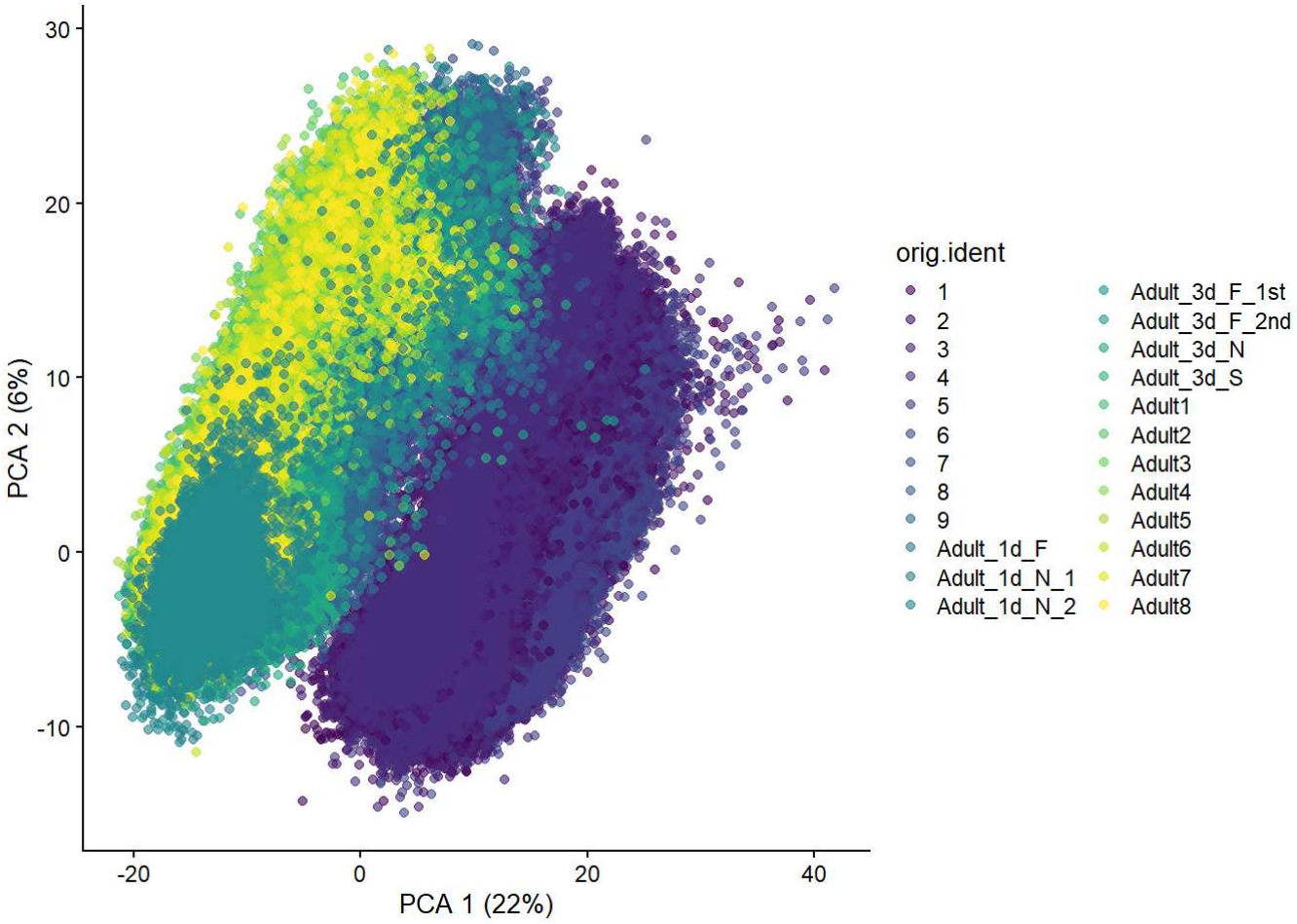
3.1 Perform library size normalization using two methods.

```
library(SingleCellExperiment)
library(scran)
library(scater)

# Copy the SingleCellExperiment object
sce_norm <- sce

# Calculate CPM normalization for sce_norm
sce_norm_cpm <- calculateCPM(sce_norm)
assays(sce_norm)$CPM <- sce_norm_cpm  # Store CPM in a new assay for sce_norm

# Log-transform the CPM values manually for sce_norm
assays(sce_norm)$logCPM <- log1p(assays(sce_norm)$CPM)  # Use log1p for log-transformation

# Proceed with scran normalization for sce_norm
clusters <- quickCluster(sce_norm)
sce_norm <- computeSumFactors(sce_norm, clusters=clusters)
sce_norm <- logNormCounts(sce_norm)  # Log-transformed normalized counts
```

```
# PCA for CPM-normalized data
sce_norm_cpm_pca <- runPCA(sce_norm, exprs_values="logCPM", ncomponents=2)

# PCA for Scran-normalized data
sce_norm_scran_pca <- runPCA(sce_norm, ncomponents=2)


# Plot PCA for CPM-normalized data
plotPCA(sce_norm_cpm_pca, colour_by = "orig.ident")
```

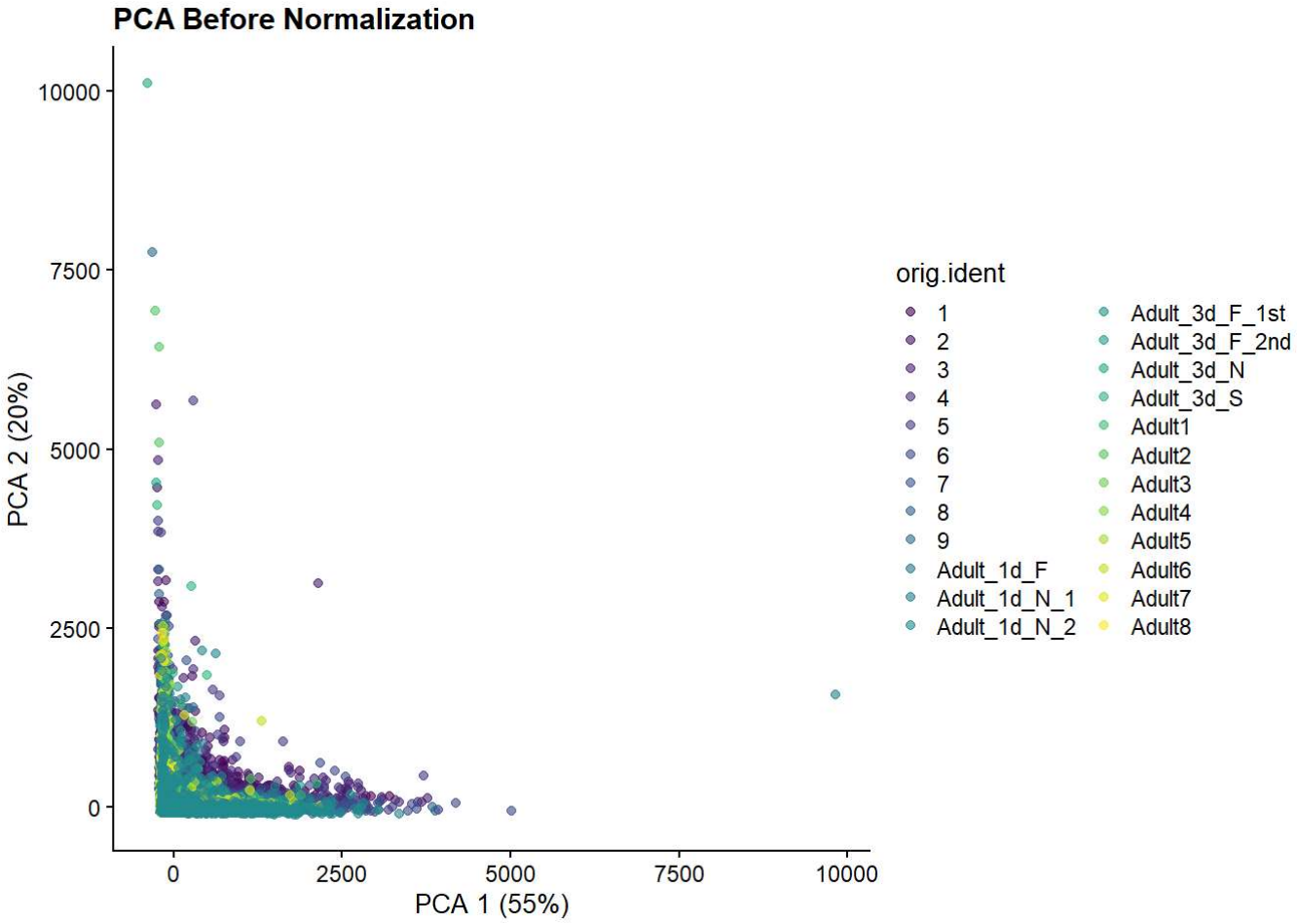



```
# Plot PCA for Scran-normalized data
plotPCA(sce_norm_scran_pca, colour_by = "orig.ident")
```

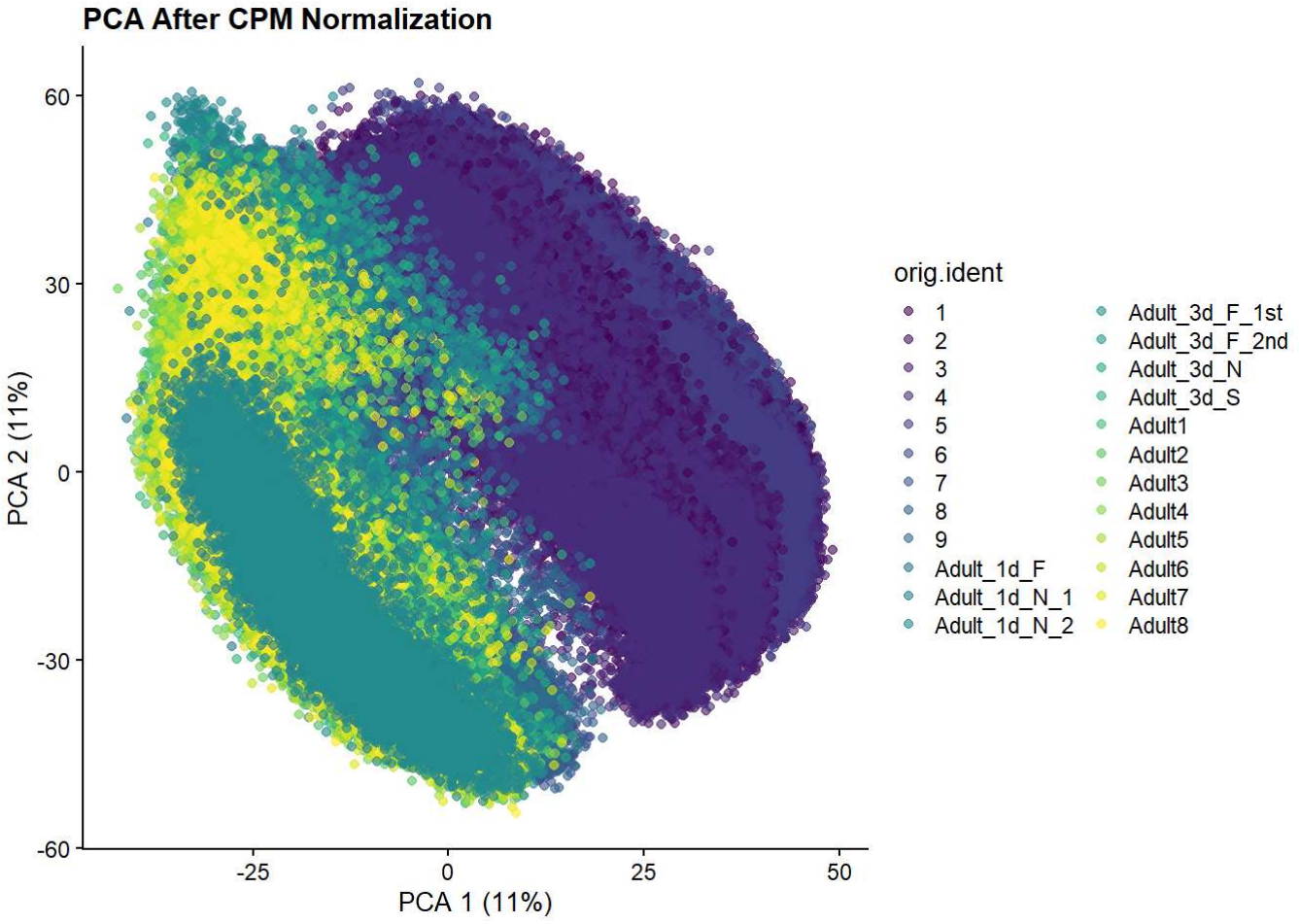


3.3 Show PCA before and after library size normalization.

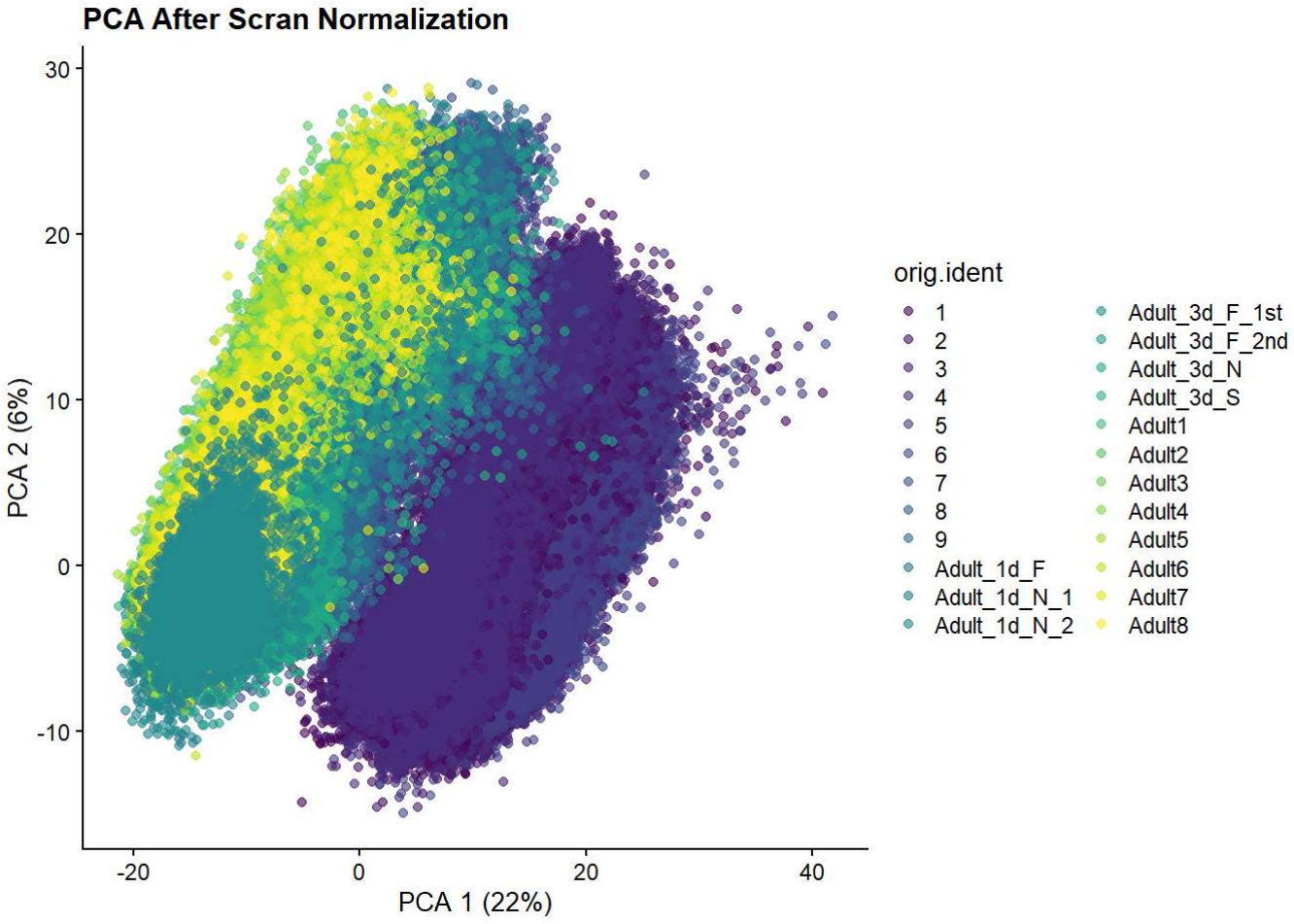
```
library(SingleCellExperiment)
library(scran)
library(scater)

# Copy the SingleCellExperiment object for raw data PCA
sce_raw <- sce

# Run PCA on the raw count data
sce_raw_pca <- runPCA(sce_raw, exprs_values = "counts", ncomponents = 2)

# Plot PCA for raw count data
plotPCA(sce_raw_pca, colour_by = "orig.ident") + ggtitle("PCA Before Normalization")
```




```
# Now plot the PCA on normalized data, which you have already performed
plotPCA(sce_norm_cpm_pca, colour_by = "orig.ident") + ggtitle("PCA After CPM Normalization")
```




```
plotPCA(sce_norm_scran_pca, colour_by = "orig.ident") + ggtitle("PCA After Scran Normalization")
```


PCA After Scran Normalization
PCA 2 (6%)
PCA 1 (22%)
30
20
10
0
-10
-20
0
20
40
orig.ident
1
2
3
4
5
6
7
8
9
Adult_1d_F
Adult_1d_N_1
Adult_1d_N_2
Adult_3d_F_1st
Adult_3d_F_2nd
Adult_3d_N
Adult_3d_S
Adult1
Adult2
Adult3
Adult4
Adult5
Adult6
Adult7
Adult8