```
library(Seurat)
library(Matrix)
library(dplyr)
library(ggplot2)

control4.data=Read10X("F:\\sample_raw_feature_bc_matrix
(1).tar\\sample_raw_feature_bc_matrix (1)")

control4=CreateSeuratObject(counts=control4.data, project="Control4")


control4$log10GenePerUMI=log10(control4$nFeature_RNA) / log10(control4$nCount_RNA)

control4$percent.mito=PercentageFeatureSet(control4,pattern='^mt-')
control4$percent.mito=control4@meta.data$percent.mito / 100

metadata=control4@meta.data
metadata$cells=rownames(metadata)
metadata=metadata %>% dplyr :: rename("seq_folder"="orig.ident", "nGene"="nFeature_RNA",
"nUMI"= "nCount_RNA")
control4@meta.data=metadata

#cell count

metadata %>%
  ggplot(aes(x=seq_folder, fill=seq_folder)) +
  geom_bar() +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  theme(plot.title = element_text(hjust=0.5, face="bold")) +
  ggtitle("NCells")

# Visualize the number UMIs/transcripts per cell
metadata %>%
  ggplot(aes(color=seq_folder, x=nUMI, fill= seq_folder)) +
  geom_density(alpha = 0.2) +
  scale_x_log10() +
  theme_classic() +
  ylab("Cell density") +
  scale_x_continuous(limits = c(0, 100))+
  geom_vline(xintercept = 500)

head(control4@meta.data, 5)
```

```
metadata %>%
  ggplot(aes(color=seq_folder, x=nGene, fill= seq_folder)) +
  geom_density(alpha = 0.2) +
  theme_classic() +
  scale_x_log10() +
  geom_vline(xintercept = 300)

# Visualize the overall complexity of the gene expression by visualizing the genes detected per
UMI (novelty score)
metadata %>%
  ggplot(aes(color=seq_folder, x=log10GenePerUMI, fill= seq_folder)) +
  geom_density(alpha = 0.2) +
  scale_x_log10() +
  theme_classic() +
  ylab("Cell density") +
  scale_x_continuous(limits = c(0.75,1))+
  geom_vline(xintercept = 0.8)

metadata %>%
  ggplot(aes(color=seq_folder, x=percent.mito, fill= seq_folder)) +
  geom_density(alpha = 0.2) +
  scale_x_log10() +
  theme_classic() +
  ylab("Cell density") +
  geom_vline(xintercept = 0.2)

metadata %>%
  ggplot(aes(x=nUMI, y=nGene, color=percent.mito)) +
  geom_point() +
  scale_color_gradient(low = "gray90", high = "black") +
  stat_smooth(method=lm) +
  scale_x_log10() +
  scale_y_log10() +
  theme_classic() +
  geom_vline(xintercept = 500) +
  geom_hline(yintercept = 300)
facet_wrap(~seq_folder)



control4 = subset(control4, subset= (nUMI>=500) & log10GenePerUMI > 0.8 & percent.mito
<0.10)
```

```
counts= GetAssayData(control4, slot="counts")
nonzero=counts>0
keep_genes=Matrix::rowSums(nonzero)>=10
filtered_counts=counts[keep_genes, ]

control4_filtered=CreateSeuratObject(counts=filtered_counts, project="Control4", meta.data =
control4@meta.data)

metadata2=control4@meta.data
metadata2$cells=rownames(metadata)
metadata2=metadata %>% dplyr :: rename("seq_folder"="orig.ident",
"nGene"="nFeature_RNA", "nUMI"= "nCount_RNA")
control4@meta.data=metadata2



metadata2 %>%
  ggplot(aes(x=seq_folder, fill=seq_folder)) +
  geom_bar() +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  theme(plot.title = element_text(hjust=0.5, face="bold")) +
  ggtitle("NCells")


# Visualize the number UMIs/transcripts per cell
metadata2 %>%
  ggplot(aes(color=seq_folder, x=nUMI, fill= seq_folder)) +
  geom_density(alpha = 0.2) +
  scale_x_log10() +
  theme_classic() +
  ylab("Cell density") +
  scale_x_continuous(limits = c(0, 100000))+
  geom_vline(xintercept = 500)



metadata2 %>%
  ggplot(aes(color=seq_folder, x=nGene, fill= seq_folder)) +
  geom_density(alpha = 0.2) +
  theme_classic() +
  scale_x_log10() +
```

```
  geom_vline(xintercept = 300)

# Visualize the overall complexity of the gene expression by visualizing the genes detected per UMI (novelty score)
metadata2 %>%
  ggplot(aes(color=seq_folder, x=log10GenePerUMI, fill= seq_folder)) +
  geom_density(alpha = 0.2) +
  scale_x_log10() +
  theme_classic() +
  ylab("Cell density") +
  scale_x_continuous(limits = c(0.75,1))+
  geom_vline(xintercept = 0.8)

metadata2 %>%
  ggplot(aes(color=seq_folder, x=percent.mito, fill= seq_folder)) +
  geom_density(alpha = 0.2) +
  scale_x_log10() +
  theme_classic() +
  ylab("Cell density") +
  geom_vline(xintercept = 0.15)

metadata2 %>%
  ggplot(aes(x=nUMI, y=nGene, color=percent.mito)) +
  geom_point() +
  scale_color_gradient(low = "gray90", high = "black") +
  stat_smooth(method=lm) +
  scale_x_log10() +
  scale_y_log10() +
  theme_classic() +
  geom_vline(xintercept = 500) +
  geom_hline(yintercept = 300)
facet_wrap(~seq_folder)


control4_filtered <- FindVariableFeatures(control4_filtered, selection.method = "vst", nfeatures = 2000)

top10 <- head(VariableFeatures(control4_filtered), 10)
plot1 <- VariableFeaturePlot(control4_filtered)
plot2 <- LabelPoints(plot = plot1, points = top10, repel = TRUE)
plot1 + plot2
```

```
control4_filtered <- NormalizeData(control4_filtered, normalization.method = "LogNormalize",
scale.factor = 10000)

all.genes <- rownames(control4_filtered)
control4_filtered <- ScaleData(control4_filtered, features = all.genes)

control4_filtered <- RunPCA(control4_filtered, features = VariableFeatures(object =
control4_filtered))

print(control4_filtered[["pca"]], dims = 1:5, nfeatures = 5)

VizDimLoadings(control4_filtered, dims = 1:2, reduction = "pca")

ElbowPlot(control4_filtered)
ElbowPlot(control4_filtered, ndims = 40)

control4_filtered<- FindNeighbors(control4_filtered, dims = 1:21)
control4_filtered <- FindClusters(control4_filtered, resolution = 0.5)


control4_filtered <- RunUMAP(control4_filtered, dims = 1:21)
DimPlot(control4_filtered, reduction = "umap")
```