SI4\_script1.single.R

Julia Steger

2022-06-29

library("Seurat")

library("Matrix")  
library("readxl")  
library("dplyr")

setup = F #this will generate the gene annotations...   
Gast18 = F  
Gast24 = F  
Gast25 = F  
Pla2d = F  
Pla3d = F  
Pla4d = F  
Pla4dc = F  
Pla5d = F  
polyp8d = F  
polyp16d = F  
phbw = F  
pha = F  
bw = F  
tentacle = F  
mes = F  
AdultMesenteryF = F  
cluster.annotation = F  
  
if (setup)  
 #gene annotations  
{  
 #load and update gene names...   
 #first, the features file from the cellranger mapping:  
 genes = read\_excel("SI3\_NVE annotations.xlsx",  
 sheet = 'cellranger.features') #this is the NVE version currently in use  
 genes <- as.data.frame(genes)  
 #\*# update for your system  
 annotations <- read\_excel("SI3\_NVE annotations.xlsx",  
 sheet = 'NVE.JGI.annotations')   
 genes<-merge(genes, annotations, by="NVE", all.x=T, sort = F) #this does work wonderfully fast  
  
 # load TFs  
 TF\_list <- read\_excel("SI3\_NVE annotations.xlsx",  
 sheet = 'TF')   
   
 #generate some gene lists for filtering:  
  
 mito.genes <- grep(pattern = "mitochondrial", genes$annotation\_notes)  
 mitochondria = genes$gene\_short\_name[mito.genes]  
   
 save.image(file = 'GenesNVE.RData')  
}  
  
if (Gast18)  
 {   
 raw.data1 <- Read10X(data.dir="~/18hr\_10000NVE")  
   
 # set the gene names to the annotations for ease of analysis  
 rownames(raw.data1) <- genes$gene\_short\_name  
   
 #calculate mitochondrial fraction   
 percent.mito1 <- Matrix::colSums(raw.data1[mitochondria, ])/Matrix::colSums(raw.data1)  
   
 #generate Seurat object   
 earlygast <- CreateSeuratObject(counts = raw.data1, project = "Gast18")   
   
 #add mitochondria information  
 earlygast[["percent.mt"]] <- PercentageFeatureSet(object = earlygast, features = mitochondria)  
   
 #add library information  
 levels(earlygast@meta.data$orig.ident) <- 'earlygast'  
   
 #filter the cells by genes detected   
 VlnPlot(earlygast, features = c('nFeature\_RNA','nCount\_RNA','percent.mt')) #Visual guide  
 earlygast <- subset(x = earlygast, subset = nFeature\_RNA > 300 & nCount\_RNA < 100000 & percent.mt < 10)  
   
 #clean up the workspace   
 rm (raw.data1)  
   
 #run standard Seurat pipeline:  
   
 #normalize  
 earlygast <- NormalizeData(earlygast, normalization.method = "LogNormalize", scale.factor = 10000)  
   
 #calculate variable genes  
 earlygast <- FindVariableFeatures(earlygast,nfeatures = 2000)  
   
 #scale and center the data  
 earlygast <- ScaleData(earlygast)  
   
 #run PCA  
 earlygast <- RunPCA(earlygast, pcs.compute = 50)  
 #evaluate standard deviations and choose number of dimensions (d)  
 ElbowPlot(object = earlygast, ndims = 30)  
 d= c(1:10)  
   
 #cluster data  
 earlygast <- FindNeighbors(earlygast, dims = 1:10, nn.method = 'annoy', annoy.metric = 'cosine')   
   
 earlygast <- FindClusters(object = earlygast,resolution = 0.2,random.seed = 0)  
 #bw <- BuildClusterTree(object = bw, reorder = TRUE,  
 # dims = d,reorder.numeric = T)  
   
 #UMAP  
 earlygast <- RunUMAP(earlygast, n.neighbors = 30,spread = 1, seed.use = 1, dims =d)  
 DimPlot(earlygast, label = T,label.size = 4, repel = F)+NoAxes()  
   
 save(earlygast, file = 'earlygast')  
   
 #PLOT LIBRARY COLOR  
 DimPlot(earlygast, label=F, label.size=6, pt.size=0.00001, cols=c("#053061", "#053061", "#053061", "#053061", "#053061", "#053061"))& NoLegend() + NoAxes()  
   
 #PLOT CLUSTER COLOR  
 DimPlot(earlygast, label=T, label.size=6, pt.size=0.5, cols=c("#A1D99B", "#E7CB94", "#9ECAE1", "#9ECAE1", "#9ECAE1", "#5254A3"))& NoLegend() + NoAxes()   
   
 }  
  
if (Gast24) {   
 #\*#direct to the matrix files of interest here:   
 raw.data1 <- Read10X(data.dir = "~/gastrula3\_24hpf\_round3")  
   
 # set the gene names to the annotations for ease of analysis  
 rownames(raw.data1) <- genes$gene\_short\_name  
   
 #calculate mitochondrial fraction   
 percent.mito1 <- Matrix::colSums(raw.data1[mitochondria, ])/Matrix::colSums(raw.data1)  
   
 #generate Seurat object   
 gast3 <- CreateSeuratObject(counts = raw.data1, project = "plalive")   
   
 #add mitochondria information  
 gast3[["percent.mt"]] <- PercentageFeatureSet(object = gast3, features = mitochondria)  
   
 #filter the cells by genes detected   
 VlnPlot(gast3, features = c('nFeature\_RNA','nCount\_RNA','percent.mt')) #Visual guide  
 gast3 <- subset(x = gast3, subset = nFeature\_RNA > 250 & nCount\_RNA < 10000 & percent.mt < 10)  
   
 #add library info to names for later identification  
 gast3 <- RenameCells(gast3, add.cell.id = "gast3")  
   
 #clean up the workspace   
 rm (raw.data1)  
   
 #run standard Seurat pipeline:  
   
 #normalize  
 gast3 <- NormalizeData(gast3, normalization.method = "LogNormalize", scale.factor = 10000)  
   
 #calculate variable genes  
 gast3 <- FindVariableFeatures(gast3,nfeatures = 2000)  
   
 #scale and center the data  
 gast3 <- ScaleData(gast3)  
   
 #run PCA  
 gast3 <- RunPCA(gast3, pcs.compute = 50)  
 #evaluate standard deviations and choose number of dimensions (d)  
 ElbowPlot(object = gast3, ndims = 50)  
 d= c(1:10)  
   
 #cluster data  
 gast3 <- FindNeighbors(gast3, dims = 1:10, nn.method = 'annoy', annoy.metric = 'cosine')   
   
 gast3 <- FindClusters(object = gast3,resolution = 0.35,random.seed = 0)  
 #bw <- BuildClusterTree(object = bw, reorder = TRUE,  
 # dims = d,reorder.numeric = T)  
   
 #UMAP  
 gast3 <- RunUMAP(gast3, n.neighbors = 30,spread = 1,seed.use = 5, dims =d)  
 DimPlot(gast3, label = T,label.size = 4, repel = F)+NoAxes()  
   
 save(gast3, file = 'gast3')  
   
 #PLOT LIBRARY COLOR  
 DimPlot(gast3, label=F, label.size=6, pt.size=0.00001, cols=c("#2166AC","#2166AC","#2166AC","#2166AC",  
 "#2166AC","#2166AC","#2166AC","#2166AC"))& NoLegend() + NoAxes()  
   
 #PLOT CLUSTER COLOR  
 DimPlot(gast3, label=T, label.size=6, pt.size=0.5, cols=c("#FDD0A2", "#E7BA52", "#FD8D3C", "#A1D99B",  
 "#9ECAE1", "#BD9E39", "#9ECAE1", "#5254A3"))& NoLegend() + NoAxes()   
}  
  
if (Gast25) {   
 #\*#direct to the matrix files of interest here:   
 raw.data1 <- Read10X(data.dir="~/gast2")  
   
 # set the gene names to the annotations for ease of analysis  
 rownames(raw.data1) <- genes$gene\_short\_name  
   
 #calculate mitochondrial fraction   
 percent.mito1 <- Matrix::colSums(raw.data1[mitochondria, ])/Matrix::colSums(raw.data1)  
   
 #generate Seurat object   
 gast25 <- CreateSeuratObject(counts = raw.data1, project = "Gast25")   
   
 #add mitochondria information  
 gast25[["percent.mt"]] <- PercentageFeatureSet(object = gast25, features = mitochondria)  
   
 #add library information  
 levels(gast25@meta.data$orig.ident) <- 'gast25'  
   
 #filter the cells by genes detected   
 VlnPlot(gast25, features = c('nFeature\_RNA','nCount\_RNA','percent.mt')) #Visual guide  
 gast25 <- subset(x = gast25, subset = nFeature\_RNA > 250 & nCount\_RNA < 30000 & percent.mt < 9)  
   
 #clean up the workspace   
 rm (raw.data1)  
   
 #run standard Seurat pipeline:  
   
 #normalize  
 gast25 <- NormalizeData(gast25, normalization.method = "LogNormalize", scale.factor = 10000)  
   
 #calculate variable genes  
 gast25 <- FindVariableFeatures(gast25,nfeatures = 2000)  
   
 #scale and center the data  
 gast25 <- ScaleData(gast25)  
   
 #run PCA  
 gast25 <- RunPCA(gast25, pcs.compute = 50)  
 #evaluate standard deviations and choose number of dimensions (d)  
 ElbowPlot(object = gast25, ndims = 20)  
 d= c(1:8)  
   
 #cluster data  
 gast25 <- FindNeighbors(gast25, dims = 1:8, nn.method = 'annoy', annoy.metric = 'cosine')   
   
 gast25 <- FindClusters(object = gast25,resolution = 0.06,random.seed = 0)  
 #bw <- BuildClusterTree(object = bw, reorder = TRUE,  
 # dims = d,reorder.numeric = T)  
   
 #UMAP  
 gast25 <- RunUMAP(gast25, n.neighbors = 30,spread = 0.5,seed.use = 1, dims =d)  
 DimPlot(gast25, label = T,label.size = 4, repel = F)+NoAxes()  
   
 save(gast25, file = 'gast25')  
   
 #PLOT LIBRARY COLOR  
 DimPlot(gast25, label=F, label.size=6, pt.size=0.00001, cols=c("#4393C3","#4393C3","#4393C3"))& NoLegend() + NoAxes()  
   
 #PLOT CLUSTER COLOR  
 DimPlot(gast25, label=T, label.size=6, pt.size=0.5, cols=c("#FD8D3C","#9ECAE1", "#A1D99B"))& NoLegend() + NoAxes()   
   
 }  
  
if (Pla2d)   
 {   
   
 #\*#direct to the matrix files of interest here:   
 raw.data <- Read10X(data.dir="~/2d\_10000NVE")  
   
 # set the gene names to the annotations for ease of analysis  
 rownames(raw.data1) <- genes$gene\_short\_name  
   
 #calculate mitochondrial fraction   
 percent.mito1 <- Matrix::colSums(raw.data1[mitochondria, ])/Matrix::colSums(raw.data1)  
   
 #generate Seurat object   
 earlypla <- CreateSeuratObject(counts = raw.data1, project = "Pla2d")   
   
 #add mitochondria information  
 gast3[["percent.mt"]] <- PercentageFeatureSet(object = gast3, features = mitochondria)  
   
 #add library information  
 levels(earlypla@meta.data$orig.ident) <- 'earlypla'  
   
 #filter the cells by genes detected   
 VlnPlot(earlypla, features = c('nFeature\_RNA','nCount\_RNA','percent.mt')) #Visual guide  
 earlypla <- subset(x = earlypla, subset = nFeature\_RNA > 300 & nCount\_RNA < 100000 & percent.mt < 10)  
   
 #clean up the workspace   
 rm (raw.data1)  
   
 #run standard Seurat pipeline:  
   
 #normalize  
 earlypla <- NormalizeData(earlypla, normalization.method = "LogNormalize", scale.factor = 10000)  
   
 #calculate variable genes  
 earlypla <- FindVariableFeatures(earlypla,nfeatures = 2000)  
   
 #scale and center the data  
 earlypla <- ScaleData(earlypla)  
   
 #run PCA  
 earlypla <- RunPCA(earlypla, pcs.compute = 50)  
 #evaluate standard deviations and choose number of dimensions (d)  
 ElbowPlot(object = earlypla, ndims = 30)  
 d= c(1:20)  
   
 #cluster data  
 earlypla <- FindNeighbors(earlypla, dims = 1:20, nn.method = 'annoy', annoy.metric = 'cosine')   
   
 earlypla <- FindClusters(object = earlypla,resolution = 0.5,random.seed = 0)  
 #bw <- BuildClusterTree(object = bw, reorder = TRUE,  
 # dims = d,reorder.numeric = T)  
   
 #UMAP  
 earlypla <- RunUMAP(earlypla, n.neighbors = 30,spread = 0.75,seed.use = 1, dims =d)  
 DimPlot(earlypla, label = T,label.size = 4, repel = F)+NoAxes()  
   
 save(earlypla, file = 'earlypla')  
   
 #PLOT LIBRARY COLOR  
 DimPlot(earlypla, label=F, label.size=6, pt.size=0.00001, cols=c("#025656", "#025656","#025656","#025656","#025656",  
 "#025656","#025656","#025656","#025656","#025656",  
 "#025656","#025656","#025656","#025656"))& NoLegend() + NoAxes()  
   
 #PLOT CLUSTER COLOR  
 DimPlot(earlypla, label=T, label.size=6, pt.size=0.5, cols=c("#FD8D3C", "#FD8D3C","#FD8D3C", "#E6550D", "#E7CB94",  
 "#FDD0A2", "#31A354", "#BD9E39", "#5254A3", "#3182BD",  
 "#3182BD", "#3182BD", "#E7CB94", "#E7BA52"))& NoLegend() + NoAxes()   
 }  
  
if (Pla3d)   
{   
 #\*#direct to the matrix files of interest here:   
 raw.data1 <- Read10X(data.dir = '~/Pla3d')  
   
 # set the gene names to the annotations for ease of analysis  
 rownames(raw.data1) <- genes$gene\_short\_name  
   
 #calculate mitochondrial fraction   
 percent.mito1 <- Matrix::colSums(raw.data1[mitochondria, ])/Matrix::colSums(raw.data1)  
   
 #generate Seurat object   
 midpla <- CreateSeuratObject(counts = raw.data1, project = "midpla")   
   
 #add mitochondria information  
 midpla[["percent.mt"]] <- PercentageFeatureSet(object = midpla, features = mitochondria)  
   
 #add library information  
 levels(midpla@meta.data$orig.ident) <- 'midpla'  
   
 #filter the cells by genes detected   
 VlnPlot(midpla, features = c('nFeature\_RNA','nCount\_RNA','percent.mt')) #Visual guide  
 midpla <- subset(x = midpla, subset = nFeature\_RNA > 250 & nCount\_RNA < 25000 & percent.mt < 10)  
   
 #clean up the workspace   
 rm (raw.data1)  
   
 #run standard Seurat pipeline:  
   
 #normalize  
 midpla <- NormalizeData(midpla, normalization.method = "LogNormalize", scale.factor = 10000)  
   
 #calculate variable genes  
 midpla <- FindVariableFeatures(midpla,nfeatures = 2000)  
   
 #scale and center the data  
 midpla <- ScaleData(midpla)  
   
 #run PCA  
 midpla <- RunPCA(midpla, pcs.compute = 50)  
 #evaluate standard deviations and choose number of dimensions (d)  
 ElbowPlot(object = midpla, ndims = 50)  
 d= c(1:10)  
   
 #cluster data  
 midpla <- FindNeighbors(midpla, dims = 1:10, nn.method = 'annoy', annoy.metric = 'cosine')   
   
 midpla <- FindClusters(object = midpla,resolution = 0.7,random.seed = 0)  
 #bw <- BuildClusterTree(object = bw, reorder = TRUE,  
 # dims = d,reorder.numeric = T)  
   
 #UMAP  
 midpla <- RunUMAP(midpla, n.neighbors = 25,spread = 0.5,seed.use = 1, dims =d)  
 DimPlot(midpla, label = T,label.size = 4, repel = F)+NoAxes()  
   
 save(midpla, file = 'midpla')  
   
 #PLOT LIBRARY COLOR  
 DimPlot(midpla, label=F, label.size=6, pt.size=0.00001, cols=c("#037272", "#037272", "#037272", "#037272",  
 "#037272", "#037272", "#037272", "#037272"))& NoLegend() + NoAxes()  
   
 #PLOT CLUSTER COLOR  
 DimPlot(midpla, label=T, label.size=6, pt.size=0.5, cols=c("#E6550D", "#E7CB94", "#FD8D3C", "#FDD0A2",  
 "#E7BA52", "#31A354", "#BD9E39", "#3182BD"))& NoLegend() + NoAxes()   
 }  
  
  
  
if (Pla4d)   
 {   
 #\*#direct to the matrix files of interest here:   
 raw.data1 <- Read10X(data.dir = '~/Nv4d')  
   
 # set the gene names to the annotations for ease of analysis  
 rownames(raw.data1) <- genes$gene\_short\_name  
   
 #calculate mitochondrial fraction   
 percent.mito1 <- Matrix::colSums(raw.data1[mitochondria, ])/Matrix::colSums(raw.data1)  
   
 #generate Seurat object   
 plalive <- CreateSeuratObject(counts = raw.data1, project = "plalive")   
   
 #add mitochondria information  
 plalive[["percent.mt"]] <- PercentageFeatureSet(object = plalive, features = mitochondria)  
   
 #add library information  
 levels(plalive@meta.data$orig.ident) <- 'plalive'  
   
 #filter the cells by genes detected   
 VlnPlot(plalive, features = c('nFeature\_RNA','nCount\_RNA','percent.mt')) #Visual guide  
 plalive <- subset(x = plalive, subset = nFeature\_RNA > 300 & nCount\_RNA < 10000 & percent.mt < 10)  
   
 #clean up the workspace   
 rm (raw.data1)  
   
 #run standard Seurat pipeline:  
   
 #normalize  
 plalive <- NormalizeData(plalive, normalization.method = "LogNormalize", scale.factor = 10000)  
   
 #calculate variable genes  
 plalive <- FindVariableFeatures(plalive,nfeatures = 2000)  
   
 #scale and center the data  
 plalive <- ScaleData(plalive)  
   
 #run PCA  
 plalive <- RunPCA(plalive, pcs.compute = 50)  
 #evaluate standard deviations and choose number of dimensions (d)  
 ElbowPlot(object = plalive, ndims = 50)  
 d= c(1:15)  
   
 #cluster data  
 plalive <- FindNeighbors(plalive, dims = 1:15, nn.method = 'annoy', annoy.metric = 'cosine')   
   
 plalive <- FindClusters(object = plalive,resolution = 0.5,random.seed = 0)  
 #bw <- BuildClusterTree(object = bw, reorder = TRUE,  
 # dims = d,reorder.numeric = T)  
   
 #UMAP  
 plalive <- RunUMAP(plalive, n.neighbors = 30,spread = 0.5,seed.use = 1, dims =d)  
 DimPlot(plalive, label = T,label.size = 4, repel = F)+NoAxes()  
   
 save(plalive, file = 'plalive')  
   
 #PLOT LIBRARY COLOR  
 DimPlot(plalive, label=F, label.size=6, pt.size=0.00001, cols=c("#00A08A", "#00A08A", "#00A08A", "#00A08A", "#00A08A",   
 "#00A08A", "#00A08A", "#00A08A", "#00A08A", "#00A08A",   
 "#00A08A", "#00A08A", "#00A08A"))& NoLegend() + NoAxes()  
 #PLOT CLUSTER COLOR  
 DimPlot(plalive, label=T, label.size=6, pt.size=0.5, cols=c("#FD8D3C", "#FD8D3C", "#FDD0A2", "#E6550D", "#E7CB94",  
 "#E7CB94", "#31A354", "#3182BD","#3182BD", "#BD9E39", "#5254A3"))& NoLegend() + NoAxes()   
 }  
  
  
  
if (Pla4dc) {  
   
 #\*#direct to the matrix files of interest here:   
 raw.data1 <- Read10X(data.dir = '~/Pla4d\_cryo')  
   
 # set the gene names to the annotations for ease of analysis  
 rownames(raw.data1) <- genes$gene\_short\_name  
   
 #calculate mitochondrial fraction   
 percent.mito1 <- Matrix::colSums(raw.data1[mitochondria, ])/Matrix::colSums(raw.data1)  
   
 #generate Seurat object   
 placryo <- CreateSeuratObject(counts = raw.data1, project = "placryo")   
   
 #add mitochondria information  
 placryo[["percent.mt"]] <- PercentageFeatureSet(object = placryo, features = mitochondria)  
   
 #add library information  
 levels(placryo@meta.data$orig.ident) <- 'placryo'  
   
 #filter the cells by genes detected   
 VlnPlot(placryo, features = c('nFeature\_RNA','nCount\_RNA','percent.mt')) #Visual guide  
 placryo <- subset(x = placryo, subset = nFeature\_RNA > 300 & nCount\_RNA < 10000 & percent.mt < 10)  
   
 #clean up the workspace   
 rm (raw.data1)  
   
 #run standard Seurat pipeline:  
   
 #normalize  
 placryo <- NormalizeData(placryo, normalization.method = "LogNormalize", scale.factor = 10000)  
   
 #calculate variable genes  
 placryo <- FindVariableFeatures(placryo,nfeatures = 2000)  
   
 #scale and center the data  
 placryo <- ScaleData(placryo)  
   
 #run PCA  
 placryo <- RunPCA(placryo, pcs.compute = 50)  
 #evaluate standard deviations and choose number of dimensions (d)  
 ElbowPlot(object = placryo, ndims = 50)  
 d= c(1:15)  
   
 #cluster data  
 placryo <- FindNeighbors(placryo, dims = 1:15, nn.method = 'annoy', annoy.metric = 'cosine')   
   
 placryo <- FindClusters(object = placryo,resolution = 0.5,random.seed = 0)  
 #bw <- BuildClusterTree(object = bw, reorder = TRUE,  
 # dims = d,reorder.numeric = T)  
   
 #UMAP  
 placryo <- RunUMAP(placryo, n.neighbors = 30,spread = 0.5,seed.use = 1, dims =d)  
 DimPlot(placryo, label = T,label.size = 4, repel = F)+NoAxes()  
   
 save(placryo, file = 'placryo')  
   
 #PLOT LIBRARY COLOR  
 DimPlot(placryo, label=F, label.size=6, pt.size=0.00001, cols=c("#66C6B8", "#66C6B8", "#66C6B8", "#66C6B8",   
 "#66C6B8", "#66C6B8", "#66C6B8", "#66C6B8"))& NoLegend() + NoAxes()  
 #PLOT CLUSTER COLOR  
 DimPlot(placryo, label=T, label.size=6, pt.size=0.5, cols=c("#FD8D3C", "#E6550D", "#E7CB94", "#3182BD",  
 "#FDD0A2", "#E7CB94", "#3182BD", "#31A354"))& NoLegend() + NoAxes()   
}  
  
  
  
if (Pla5d)   
{   
 #\*#direct to the matrix files of interest here:   
 raw.data1 <- Read10X(data.dir = '~/Pla5d')  
   
 # set the gene names to the annotations for ease of analysis  
 rownames(raw.data1) <- genes$gene\_short\_name  
   
 #calculate mitochondrial fraction   
 percent.mito1 <- Matrix::colSums(raw.data1[mitochondria, ])/Matrix::colSums(raw.data1)  
   
 #generate Seurat object   
 tentbud <- CreateSeuratObject(counts = raw.data1, project = "tentbud")   
   
 #add mitochondria information  
 tentbud[["percent.mt"]] <- PercentageFeatureSet(object = tentbud, features = mitochondria)  
   
 #add library information  
 levels(tentbud@meta.data$orig.ident) <- 'tentbud'  
   
 #filter the cells by genes detected   
 VlnPlot(tentbud, features = c('nFeature\_RNA','nCount\_RNA','percent.mt')) #Visual guide  
 tentbud <- subset(x = tentbud, subset = nFeature\_RNA > 250 & nCount\_RNA < 20000 & percent.mt < 10)  
   
 #clean up the workspace   
 rm (raw.data1)  
   
 #run standard Seurat pipeline:  
   
 #normalize  
 tentbud <- NormalizeData(tentbud, normalization.method = "LogNormalize", scale.factor = 10000)  
   
 #calculate variable genes  
 tentbud <- FindVariableFeatures(tentbud,nfeatures = 2000)  
   
 #scale and center the data  
 tentbud <- ScaleData(tentbud)  
   
 #run PCA  
 tentbud <- RunPCA(tentbud, pcs.compute = 50)  
 #evaluate standard deviations and choose number of dimensions (d)  
 ElbowPlot(object = tentbud, ndims = 50)  
 d= c(1:20)  
   
 #cluster data  
 tentbud <- FindNeighbors(tentbud, dims = 1:20, nn.method = 'annoy', annoy.metric = 'cosine')   
   
 tentbud <- FindClusters(object = tentbud,resolution = 0.8,random.seed = 0)  
 #bw <- BuildClusterTree(object = bw, reorder = TRUE,  
 # dims = d,reorder.numeric = T)  
   
 #UMAP  
 tentbud <- RunUMAP(tentbud, n.neighbors = 30,spread = 0.5,seed.use = 1, dims =d)  
 DimPlot(tentbud, label = T,label.size = 4, repel = F)+NoAxes()  
   
 save(tentbud, file = 'tentbud')  
   
 #PLOT LIBRARY COLOR  
 DimPlot(tentbud, label=F, label.size=6, pt.size=0.00001, cols=c("#F98400","#F98400","#F98400","#F98400","#F98400",  
 "#F98400","#F98400","#F98400","#F98400","#F98400",  
 "#F98400","#F98400","#F98400","#F98400","#F98400",  
 "#F98400","#F98400","#F98400", "#F98400","#F98400",  
 "#F98400","#F98400","#F98400"))& NoLegend() + NoAxes()  
   
 #PLOT CLUSTER COLOR  
 DimPlot(tentbud, label=T, label.size=6, pt.size=0.5, cols=c("#FD8D3C","#E7BA52", "#FD8D3C", "#8C6D31", "#E6550D",  
 "#E7CB94", "#E7CB94", "#FDD0A2", "#31A354", "#E7BA52",  
 "#E7BA52", "#5254A3", "#3182BD", "#BD9E39", "#3182BD",  
 "#3182BD", "#E7CB94", "#E7CB94"))& NoLegend() + NoAxes()   
 }  
  
  
  
if (polyp8d)   
 {  
 #\*#direct to the matrix files of interest here:   
 raw.data1 <- Read10X(data.dir = '~/polyp8d')  
   
 # set the gene names to the annotations for ease of analysis  
 rownames(raw.data1) <- genes$gene\_short\_name  
   
 #calculate mitochondrial fraction   
 percent.mito1 <- Matrix::colSums(raw.data1[mitochondria, ])/Matrix::colSums(raw.data1)  
   
 #generate Seurat object   
 pol12 <- CreateSeuratObject(counts = raw.data1, project = "pol12")   
   
 #add mitochondria information  
 pol12[["percent.mt"]] <- PercentageFeatureSet(object = pol12, features = mitochondria)  
   
 #add library information  
 levels(pol12@meta.data$orig.ident) <- 'pol12'  
   
 #filter the cells by genes detected   
 VlnPlot(pol12, features = c('nFeature\_RNA','nCount\_RNA','percent.mt')) #Visual guide  
 pol12 <- subset(x = pol12, subset = nFeature\_RNA > 250 & nCount\_RNA < 15000 & percent.mt < 10)  
   
 #clean up the workspace   
 rm (raw.data1)  
   
 #run standard Seurat pipeline:  
   
 #normalize  
 pol12 <- NormalizeData(pol12, normalization.method = "LogNormalize", scale.factor = 10000)  
   
 #calculate variable genes  
 pol12 <- FindVariableFeatures(pol12,nfeatures = 2000)  
   
 #scale and center the data  
 pol12 <- ScaleData(pol12)  
   
 #run PCA  
 pol12 <- RunPCA(pol12, pcs.compute = 50)  
 #evaluate standard deviations and choose number of dimensions (d)  
 ElbowPlot(object = pol12, ndims = 50)  
 d= c(1:20)  
   
 #cluster data  
 pol12 <- FindNeighbors(pol12, dims = 1:20, nn.method = 'annoy', annoy.metric = 'cosine')   
   
 pol12 <- FindClusters(object = pol12,resolution = 0.5,random.seed = 0)  
 #bw <- BuildClusterTree(object = bw, reorder = TRUE,  
 # dims = d,reorder.numeric = T)  
   
 #UMAP  
 pol12 <- RunUMAP(pol12, n.neighbors = 30,spread = 0.7,seed.use = 1, dims =d)  
 DimPlot(pol12, label = T,label.size = 4, repel = F)+NoAxes()  
   
 save(pol12, file = 'pol12')  
   
 #PLOT LIBRARY COLOR  
 DimPlot(pol12, label=F, label.size=6, pt.size=0.00001, cols=c("#F2AD00","#F2AD00","#F2AD00","#F2AD00","#F2AD00",  
 "#F2AD00","#F2AD00","#F2AD00","#F2AD00","#F2AD00",  
 "#F2AD00","#F2AD00","#F2AD00","#F2AD00","#F2AD00"))& NoLegend() + NoAxes()  
   
 #PLOT CLUSTER COLOR  
 DimPlot(pol12, label=T, label.size=6, pt.size=0.5, cols=c("#FD8D3C", "#E7CB94", "#8C6D31", "#E6550D", "#FDD0A2",  
 "#31A354", "#E7BA52", "#3182BD", "#3182BD", "#5254A3",  
 "#E7CB94","#E7CB94"))& NoLegend() + NoAxes()   
 }  
  
  
  
if (polyp16d) {   
   
 #\*#direct to the matrix files of interest here:   
 raw.data1 <- Read10X(data.dir = '~/polyp16d')  
   
 # set the gene names to the annotations for ease of analysis  
 rownames(raw.data1) <- genes$gene\_short\_name  
   
 #calculate mitochondrial fraction   
 percent.mito1 <- Matrix::colSums(raw.data1[mitochondria, ])/Matrix::colSums(raw.data1)  
   
 #generate Seurat object   
 pol3 <- CreateSeuratObject(counts = raw.data1, project = "pol3")   
   
 #add mitochondria information  
 pol3[["percent.mt"]] <- PercentageFeatureSet(object = pol3, features = mitochondria)  
   
 #add library information  
 levels(pol3@meta.data$orig.ident) <- 'pol3'  
   
 #filter the cells by genes detected   
 VlnPlot(pol3, features = c('nFeature\_RNA','nCount\_RNA','percent.mt')) #Visual guide  
 pol3 <- subset(x = pol3, subset = nFeature\_RNA > 250 & nCount\_RNA < 20000 & percent.mt < 10)  
   
 #clean up the workspace   
 rm (raw.data1)  
   
 #run standard Seurat pipeline:  
   
 #normalize  
 pol3 <- NormalizeData(pol3, normalization.method = "LogNormalize", scale.factor = 10000)  
   
 #calculate variable genes  
 pol3 <- FindVariableFeatures(pol3,nfeatures = 2000)  
   
 #scale and center the data  
 pol3 <- ScaleData(pol3)  
   
 #run PCA  
 pol3 <- RunPCA(pol3, pcs.compute = 50)  
 #evaluate standard deviations and choose number of dimensions (d)  
 ElbowPlot(object = pol3, ndims = 50)  
 d= c(1:15)  
   
 #cluster data  
 pol3 <- FindNeighbors(pol3, dims = 1:15, nn.method = 'annoy', annoy.metric = 'cosine')   
   
 pol3 <- FindClusters(object = pol3,resolution = 0.8,random.seed = 0)  
 #bw <- BuildClusterTree(object = bw, reorder = TRUE,  
 # dims = d,reorder.numeric = T)  
   
 #UMAP  
 pol3 <- RunUMAP(pol3, n.neighbors = 30,spread = 0.5,seed.use = 1, dims =d)  
 DimPlot(pol3, label = T,label.size = 4, repel = F)+NoAxes()  
   
 save(pol3, file = 'pol3')  
   
 #PLOT LIBRARY COLOR  
 DimPlot(pol3, label=F, label.size=6, pt.size=0.00001, cols=c("#E2D200","#E2D200","#E2D200","#E2D200","#E2D200",  
 "#E2D200","#E2D200","#E2D200","#E2D200","#E2D200",  
 "#E2D200","#E2D200","#E2D200","#E2D200","#E2D200"))& NoLegend() + NoAxes()  
   
 #PLOT CLUSTER COLOR  
 DimPlot(pol3, label=T, label.size=6, pt.size=0.5, cols=c("#FD8D3C", "#E7CB94", "#8C6D31", "#31A354", "#E6550D",  
 "#E7BA52", "#3182BD", "#5254A3", "#FDD0A2", "#E7CB94",  
 "#E7CB94", "#E7CB94"))& NoLegend() + NoAxes()   
 }  
  
  
  
  
if (phbw)   
{   
 #\*#direct to the matrix files of interest here:   
 raw.data1 <- Read10X(data.dir = '~/phbw')  
   
 # set the gene names to the annotations for ease of analysis  
 rownames(raw.data1) <- genes$gene\_short\_name  
   
 #calculate mitochondrial fraction   
 percent.mito1 <- Matrix::colSums(raw.data1[mitochondria, ])/Matrix::colSums(raw.data1)  
   
 #generate Seurat object   
 phbw <- CreateSeuratObject(counts = raw.data1, project = "phbw")   
   
 #add mitochondria information  
 phbw[["percent.mt"]] <- PercentageFeatureSet(object = bw, features = mitochondria)  
   
 #add library information  
 levels(phbw@meta.data$orig.ident) <- 'phbw'  
   
 #filter the cells by genes detected   
 VlnPlot(phbw, features = c('nFeature\_RNA','nCount\_RNA','percent.mt')) #Visual guide  
 phbw <- subset(x = phbw, subset = nFeature\_RNA > 250 & nCount\_RNA < 20000 & percent.mt < 10)  
   
 #clean up the workspace   
 rm (raw.data1)  
   
 #run standard Seurat pipeline:  
   
 #normalize  
 phbw <- NormalizeData(phbw, normalization.method = "LogNormalize", scale.factor = 10000)  
   
 #calculate variable genes  
 phbw <- FindVariableFeatures(phbw,nfeatures = 2000)  
   
 #scale and center the data  
 phbw <- ScaleData(phbw)  
   
 #run PCA  
 phbw <- RunPCA(phbw, pcs.compute = 50)  
 #evaluate standard deviations and choose number of dimensions (d)  
 ElbowPlot(object = phbw, ndims = 50)  
 d= c(1:20)  
   
 #cluster data  
 phbw <- FindNeighbors(phbw, dims = 1:20, nn.method = 'annoy', annoy.metric = 'cosine')   
   
 phbw <- FindClusters(object = phbw,resolution = 0.8,random.seed = 0)  
 #bw <- BuildClusterTree(object = bw, reorder = TRUE,  
 # dims = d,reorder.numeric = T)  
   
 #UMAP  
 phbw <- RunUMAP(phbw, n.neighbors = 30,spread = 1,seed.use = 1, dims =d)  
 DimPlot(phbw, label = T,label.size = 4, repel = F)+NoAxes()  
   
 save(phbw, file = 'bw')  
   
 #PLOT LIBRARY COLOR  
 DimPlot(phbw, label=F, label.size=6, pt.size=0.00001, cols=c("#AD2323","#AD2323","#AD2323","#AD2323","#AD2323",  
 "#AD2323","#AD2323","#AD2323","#AD2323","#AD2323",  
 "#AD2323","#AD2323","#AD2323","#AD2323"))& NoLegend() + NoAxes()  
   
 #PLOT CLUSTER COLOR  
 DimPlot(phbw, label=T, label.size=6, pt.size=0.5, cols=c("#FD8D3C", "#8C6D31", "#FDD0A2","#E7CB94", "#E7CB94",  
 "#8C6D31", "#E7CB94", "#31A354", "#3182BD", "#BD9E39",  
 "#5254A3", "#5254A3", "#E7BA52", "#E7BA52"))& NoLegend() + NoAxes()   
 }  
  
  
  
if (pha) {   
   
 #\*#direct to the matrix files of interest here:   
 raw.data1 <- Read10X(data.dir = '~/pha')  
   
 # set the gene names to the annotations for ease of analysis  
 rownames(raw.data1) <- genes$gene\_short\_name  
   
 #calculate mitochondrial fraction   
 percent.mito1 <- Matrix::colSums(raw.data1[mitochondria, ])/Matrix::colSums(raw.data1)  
   
 #generate Seurat object   
 pha <- CreateSeuratObject(counts = raw.data1, project = "pha")   
   
 #add mitochondria information  
 pha[["percent.mt"]] <- PercentageFeatureSet(object = mes, features = mitochondria)  
   
 #add library information  
 levels(pha@meta.data$orig.ident) <- 'pha'  
   
 #filter the cells by genes detected   
 VlnPlot(pha, features = c('nFeature\_RNA','nCount\_RNA','percent.mt')) #Visual guide  
 pha <- subset(x = pha, subset = nFeature\_RNA > 250 & nCount\_RNA < 10000 & percent.mt < 10)  
  
 #clean up the workspace   
 rm (raw.data1)  
   
 #run standard Seurat pipeline:  
   
 #normalize  
 pha <- NormalizeData(pha, normalization.method = "LogNormalize", scale.factor = 10000)  
   
 #calculate variable genes  
 pha <- FindVariableFeatures(pha,nfeatures = 2000)  
   
 #scale and center the data  
 pha <- ScaleData(pha)  
   
 #run PCA  
 pha <- RunPCA(pha, pcs.compute = 50)  
 #evaluate standard deviations and choose number of dimensions (d)  
 ElbowPlot(object = pha, ndims = 50)  
 d= c(1:20)  
   
 #cluster data  
 pha <- FindNeighbors(pha, dims = 1:20, nn.method = 'annoy', annoy.metric = 'cosine')   
   
 pha <- FindClusters(object = pha,resolution = 1,random.seed = 0)  
 #pha <- BuildClusterTree(object = pha, reorder = TRUE,  
 # dims = d,reorder.numeric = T)  
   
 #UMAP  
 pha <- RunUMAP(pha, n.neighbors = 25,spread = 0.5,seed.use = 1, dims =d)  
 DimPlot(pha, label = T,label.size = 4, repel = F,#group.by = 'IDs',  
 order=(levels(pha@active.ident)))+NoAxes()  
   
 save(pha, file = 'pha.Robj')  
   
   
 #PLOT LIBRARY COLOR  
 DimPlot(pha, label=F, label.size=6, pt.size=0.00001, cols=c("#FF0000", "#FF0000","#FF0000","#FF0000","#FF0000",  
 "#FF0000","#FF0000","#FF0000","#FF0000","#FF0000",  
 "#FF0000","#FF0000","#FF0000","#FF0000","#FF0000",  
 "#FF0000","#FF0000","#FF0000","#FF0000","#FF0000"  
 ))& NoLegend() + NoAxes()  
   
   
 #PLOT CLUSTER COLOR  
 DimPlot(pha, label=T, label.size=6, pt.size=0.5, cols=c("#FD8D3C", "#E7CB94", "#8C6D31", "#31A354", "#FDD0A2",  
 "#31A354", "#3182BD", "#E7CB94", "#E7BA52", "#31A354",  
 "#3182BD", "#3182BD", "#5254A3", "#5254A3"))& NoLegend() + NoAxes()  
 }  
  
  
  
if (bw)   
 {   
 #\*#direct to the matrix files of interest here:   
 raw.data1 <- Read10X(data.dir = '~/bw')  
   
 # set the gene names to the annotations for ease of analysis  
 rownames(raw.data1) <- genes$gene\_short\_name  
   
 #calculate mitochondrial fraction   
 percent.mito1 <- Matrix::colSums(raw.data1[mitochondria, ])/Matrix::colSums(raw.data1)  
   
 #generate Seurat object   
 bw <- CreateSeuratObject(counts = raw.data1, project = "bw")   
   
 #add mitochondria information  
 bw[["percent.mt"]] <- PercentageFeatureSet(object = bw, features = mitochondria)  
   
 #add library information  
 levels(bw@meta.data$orig.ident) <- 'bw'  
   
 #filter the cells by genes detected   
 VlnPlot(bw, features = c('nFeature\_RNA','nCount\_RNA','percent.mt')) #Visual guide  
 bw <- subset(x = bw, subset = nFeature\_RNA > 250 & nCount\_RNA < 5000 & percent.mt < 10)  
   
 #clean up the workspace   
 rm (raw.data1)  
   
 #run standard Seurat pipeline:  
   
 #normalize  
 bw <- NormalizeData(bw, normalization.method = "LogNormalize", scale.factor = 10000)  
   
 #calculate variable genes  
 bw <- FindVariableFeatures(bw,nfeatures = 2000)  
   
 #scale and center the data  
 bw <- ScaleData(bw)  
   
 #run PCA  
 bw <- RunPCA(bw, pcs.compute = 50)  
 #evaluate standard deviations and choose number of dimensions (d)  
 ElbowPlot(object = bw, ndims = 50)  
 d= c(1:20)  
   
 #cluster data  
 bw <- FindNeighbors(bw, dims = 1:20, nn.method = 'annoy', annoy.metric = 'cosine')   
   
 bw <- FindClusters(object = bw,resolution = 0.6,random.seed = 0)  
 #bw <- BuildClusterTree(object = bw, reorder = TRUE,  
 # dims = d,reorder.numeric = T)  
   
 #UMAP  
 bw <- RunUMAP(bw, n.neighbors = 30,spread = 0.5,seed.use = 42, dims =d)  
 DimPlot(bw, label = T,label.size = 4, repel = F)+NoAxes()  
   
 save(bw, file = 'bw')  
   
 #PLOT LIBRARY COLOR  
 DimPlot(bw, label=F, label.size=6, pt.size=0.00001, cols=c("#9B51B4", "#9B51B4", "#9B51B4", "#9B51B4", "#9B51B4",   
 "#9B51B4", "#9B51B4", "#9B51B4", "#9B51B4", "#9B51B4"))& NoLegend() + NoAxes()  
   
 #PLOT CLUSTER COLOR  
 DimPlot(bw, label=T, label.size=6, pt.size=0.5, cols=c("#31A354", "#E7CB94", "#FDD0A2", "#E6550D", "#31A354",  
 "#31A354", "#E7BA52", "#3182BD","#3182BD","#3182BD"))& NoLegend() + NoAxes()  
  
 }  
  
  
if (tentacle)   
 {   
 #\*#direct to the matrix files of interest here:   
 raw.data1 <- Read10X(data.dir = '~/tent')  
   
 # set the gene names to the annotations for ease of analysis  
 rownames(raw.data1) <- genes$gene\_short\_name  
   
 #calculate mitochondrial fraction   
 percent.mito1 <- Matrix::colSums(raw.data1[mitochondria, ])/Matrix::colSums(raw.data1)  
   
 #generate Seurat object   
 tent <- CreateSeuratObject(counts = raw.data1, project = "tent")   
   
 #add mitochondria information  
 tent[["percent.mt"]] <- PercentageFeatureSet(object = tent, features = mitochondria)  
   
 #add library information  
 levels(tent@meta.data$orig.ident) <- 'tent'  
   
 #filter the cells by genes detected   
 VlnPlot(tent, features = c('nFeature\_RNA','nCount\_RNA','percent.mt')) #Visual guide  
 tent <- subset(x = tent, subset = nFeature\_RNA > 250 & nCount\_RNA < 10000 & percent.mt < 10)  
   
 #clean up the workspace   
 rm (raw.data1)  
   
 #run standard Seurat pipeline:  
   
 #normalize  
 tent <- NormalizeData(tent, normalization.method = "LogNormalize", scale.factor = 10000)  
   
 #calculate variable genes  
 tent <- FindVariableFeatures(tent,nfeatures = 2000)  
   
 #scale and center the data  
 tent <- ScaleData(tent)  
   
 #run PCA  
 tent <- RunPCA(tent, pcs.compute = 50)  
 #evaluate standard deviations and choose number of dimensions (d)  
 ElbowPlot(object = tent, ndims = 50)  
 d= c(1:20)  
   
 #cluster data  
 tent <- FindNeighbors(tent, dims = 1:20, nn.method = 'annoy', annoy.metric = 'cosine')   
   
 tent <- FindClusters(object = tent,resolution = 1.1,random.seed = 0)  
 #tent <- BuildClusterTree(object = tent, reorder = TRUE,  
 # dims = d,reorder.numeric = T)  
   
 #UMAP  
 tent <- RunUMAP(tent, n.neighbors = 20,spread = 0.4,seed.use = 0, dims =d)  
 DimPlot(tent, label = T,label.size = 4, repel = F)+NoAxes()  
   
  
 save(tent, file = 'tent')  
   
 #PLOT LIBRARY COLOR  
 DimPlot(tent, label=F, label.size=6, pt.size=0.00001, cols=c("#530C6B","#530C6B","#530C6B","#530C6B",  
 "#530C6B","#530C6B","#530C6B","#530C6B","#530C6B",  
 "#530C6B","#530C6B","#530C6B","#530C6B"))& NoLegend() + NoAxes()  
   
   
 #PLOT CLUSTER COLOR  
 DimPlot(tent, label=T, label.size=6, pt.size=0.5, cols=c("#FD8D3C", "#31A354", "#8C6D31", "#E7BA52", "#3182BD",  
 "#3182BD", "#FD8D3C", "#FDD0A2", "#E7CB94", "#E6550D",  
 "#FD8D3C","#FD8D3C","#FD8D3C"))& NoLegend() + NoAxes()  
 }  
  
  
  
  
if (mes)   
{   
 #\*#direct to the matrix files of interest here:   
 raw.data1 <- Read10X(data.dir = 'Z~/mes')  
   
 # set the gene names to the annotations for ease of analysis  
 rownames(raw.data1) <- genes$gene\_short\_name  
   
 #calculate mitochondrial fraction   
 percent.mito1 <- Matrix::colSums(raw.data1[mitochondria, ])/Matrix::colSums(raw.data1)  
   
 #generate Seurat object   
 mes <- CreateSeuratObject(counts = raw.data1, project = "mes")   
   
 #add mitochondria information  
 mes[["percent.mt"]] <- PercentageFeatureSet(object = mes, features = mitochondria)  
   
 #add library information  
 levels(mes@meta.data$orig.ident) <- 'mes'  
   
 #filter the cells by genes detected   
 VlnPlot(mes, features = c('nFeature\_RNA','nCount\_RNA','percent.mt')) #Visual guide  
 mes <- subset(x = mes, subset = nFeature\_RNA > 250 & nCount\_RNA < 15000 & percent.mt < 10)  
  
 #clean up the workspace   
 rm (raw.data1)  
   
 #run standard Seurat pipeline:  
   
 #normalize  
 mes <- NormalizeData(mes, normalization.method = "LogNormalize", scale.factor = 10000)  
   
 #calculate variable genes  
 mes <- FindVariableFeatures(mes,nfeatures = 2000)  
   
 #scale and center the data  
 mes <- ScaleData(mes)  
   
 #run PCA  
 mes <- RunPCA(mes, pcs.compute = 50)  
 #evaluate standard deviations and choose number of dimensions (d)  
 ElbowPlot(object = mes, ndims = 50)  
 d= c(1:20)  
   
 #cluster data  
 mes <- FindNeighbors(mes, dims = 1:20, nn.method = 'annoy', annoy.metric = 'cosine')   
   
 mes <- FindClusters(object = mes,resolution = 0.9,random.seed = 0)  
 #mes <- BuildClusterTree(object = mes, reorder = TRUE,  
 # dims = d,reorder.numeric = T)  
   
 #UMAP  
 mes <- RunUMAP(mes, n.neighbors = 15,spread = 0.5,seed.use = 42, dims =d)  
 DimPlot(mes, label = T,label.size = 4, repel = F,#group.by = 'IDs',  
 order=(levels(mes@active.ident)))+NoAxes()  
   
   
 save(mes, file = 'mes.Robj')  
   
   
 #PLOT LIBRARY COLOR  
 DimPlot(mes, label=F, label.size=6, pt.size=0.00001, cols=c("#DF6FA0","#DF6FA0","#DF6FA0","#DF6FA0",  
 "#DF6FA0","#DF6FA0","#DF6FA0","#DF6FA0"))& NoLegend() + NoAxes()  
   
   
 #PLOT CLUSTER COLOR  
 DimPlot(mes, label=T, label.size=6, pt.size=0.25, cols=c("#FD8D3C", "#E7CB94", "#E7CB94", "#E7BA52",   
 "#31A354", "#5254A3", "#8C6D31", "#31A354"))& NoLegend() + NoAxes()  
 }  
  
  
  
if (AdultMesenteryF)  
{  
 #\*#direct to the matrix files of interest here:   
 raw.data1 <- Read10X(data.dir = '~/MesenteryFemale')  
  
 # set the gene names to the annotations for ease of analysis  
 rownames(raw.data1) <- genes$gene\_short\_name  
  
 #calculate mitochondrial fraction   
 percent.mito1 <- Matrix::colSums(raw.data1[mitochondria, ])/Matrix::colSums(raw.data1)  
  
 #generate Seurat object   
 mesF <- CreateSeuratObject(counts = raw.data1, project = "mesF")   
  
 #add mitochondria information  
 mesF[["percent.mt"]] <- PercentageFeatureSet(object = mesF, features = mitochondria)  
  
 #add library information  
 levels(mesF@meta.data$orig.ident) <- 'mesF'  
   
 #filter the cells by genes detected   
 VlnPlot(mesF, features = c('nFeature\_RNA','nCount\_RNA','percent.mt')) #Visual guide  
 mesF <- subset(x = mesF, subset = nFeature\_RNA > 200 & nCount\_RNA < 20000) #& percent.mt < 0.8  
 #can also filter for mitochondial fraction: high levels could indicate poor samples   
   
 #clean up the workspace   
 rm (raw.data1)  
   
 #run standard Seurat pipeline:  
 #calculate variable genes  
 mesF <- FindVariableFeatures(mesF,nfeatures = 2000)  
   
 #scale and center the data  
 mesF <- ScaleData(mesF)  
   
 #run PCA  
 mesF <- RunPCA(mesF, pcs.compute = 50)  
 #evaluate standard deviations and choose number of dimensions (d)  
 ElbowPlot(object = mesF, ndims = 50)  
 d= c(1:23)  
 #cluster data  
 mesF <- FindNeighbors(object = mesF,reduction ="pca",dims = d,  
 nn.method = 'annoy',  
 annoy.metric = 'cosine',  
 k.param = 10)  
   
 mesF <- FindClusters(object = mesF,resolution = 0.2,random.seed = 0)  
 mesF <- BuildClusterTree(object = mesF, reorder = TRUE,  
 dims = d,reorder.numeric = T)  
   
 #UMAP  
 mesF <- RunUMAP(mesF, dims = d,  
 reduction = 'pca',  
 reduction.name ='umap',reduction.key ='umap',   
 n.neighbors = 10L,   
 spread =1,   
 min.dist = 0.3,  
 local.connectivity = 100)  
 DimPlot(mesF, label = T,label.size = 4, repel = T,#group.by = 'IDs',  
 order=(levels(mesF@active.ident)))+NoAxes()  
  
 save(mesF, file = 'FemaleMes.Robj')  
   
 #PLOT LIBRARY COLOR  
 DimPlot(mesF, label=F, label.size=6, pt.size=0.00001, cols=c('pink', 'pink', 'pink', 'pink', 'pink',   
 'pink', 'pink', 'pink', 'pink', 'pink',   
 'pink'))& NoLegend() + NoAxes()  
}  
  
  
  
if (cluster.annotation)  
{   
####AUTOMATED CLUSTER ANNOTATION  
  
#select respective sheet from excel workbook  
clusternames = read\_excel("SI2b\_single\_libraries\_auto\_anno\_marker.xlsx", sheet="xxx")  
  
#CHECKPOINT  
goi = clusternames$gene\_short\_name  
goi  
  
#DotPlot  
DotPlot(tent,'RNA',features = goi)+RotatedAxis()   
  
#how to use this to assign the ID...  
tent<- BuildClusterTree(tent, dims = c(1:30),reorder = T, reorder.numeric = T)  
#assign cluster ID to the individual libraries  
tent<-ScaleData(tent,features = goi, split.by = 'orig.ident')  
cl <-length(levels(tent@active.ident))  
C.suffix <-seq(1:cl)  
  
g=length(goi)  
clName = vector()  
m=matrix(0L,g,cl)  
for (j in 1:cl)  
{  
 for (i in 1:g)  
 m[i,j]=mean(tent@assays$RNA@scale.data[goi[i],WhichCells(tent,idents = C.suffix[j])])  
 clName[j]=as.integer(which.max(m[,j]))  
}  
levels(tent@active.ident) = clusternames$label[clName]  
DimPlot(tent,label = T, pt.size=0.5, label.size=6)+NoAxes()  
 }