Supplementary File 1

```
### Read in libraries
library(Seurat)
library(ggplot2)
library(viridis)
library(RColorBrewer)
library(tidyr)
library(dplyr)
library(celda)
library(scater)
library(cowplot)
library(DoubletFinder)

##set working directory
setwd()

##Load filtered data from cellranger for preliminary analysis
counts = Seurat::Read10X(data.dir = 'Filtered')

##Run quick qc and clustering
seurat <- CreateSeuratObject(counts = counts) ##create seurat object

####Add metrics
seurat[["percent.mt"]] <- PercentageFeatureSet(seurat, pattern = "^MT-") ##Add percentage of mito reads
RPS.genes <- grep(pattern = "^RPS", x = rownames(x = seurat), value = TRUE) ##Extract ribosomal RPS genes
RPL.genes <- grep(pattern = "^RPL", x = rownames(x = seurat), value = TRUE) ##Extract ribosomal RPL genes
ribo <- c(RPS.genes, RPL.genes) ##make vector of all ribosomal genes
seurat[["percent.ribo"]] <- PercentageFeatureSet(seurat, features = ribo) ##Add as percentage feature set
```

```
seurat@meta.data$Complexity <- log10(seurat@meta.data$nFeature_RNA) /
log10(seurat@meta.data$nCount_RNA) ##Add complexity

head(seurat@meta.data)
seurat

##Plot metrics
VlnPlot(seurat, features = c("nFeature_RNA", "percent.mt", "percent.ribo","Complexity"), ncol = 4)

meta <- seurat@meta.data
meta %>% summarize(mean = mean(nFeature_RNA, na.rm = TRUE),
            median = median(nFeature_RNA, na.rm = TRUE),
            min = min(nFeature_RNA, na.rm = TRUE),
            max = max(nFeature_RNA, na.rm = TRUE),
            range = diff(range(nFeature_RNA, na.rm = TRUE)),
            quantile = list(quantile(nFeature_RNA, probs = c(0.25, 0.5, 0.75), na.rm = TRUE))) %>%
  unnest_wider(quantile)

meta <- seurat@meta.data
meta %>% summarize(mean = mean(nCount_RNA, na.rm = TRUE),
            median = median(nCount_RNA, na.rm = TRUE),
            min = min(nCount_RNA, na.rm = TRUE),
            max = max(nCount_RNA, na.rm = TRUE),
            range = diff(range(nCount_RNA, na.rm = TRUE)),
            quantile = list(quantile(nCount_RNA, probs = c(0.25, 0.5, 0.75), na.rm = TRUE))) %>%
  unnest_wider(quantile)

meta <- seurat@meta.data
meta %>% summarize(mean = mean(percent.mt, na.rm = TRUE),
            median = median(percent.mt, na.rm = TRUE),
```

```r
            min = min(percent.mt, na.rm = TRUE),
            max = max(percent.mt, na.rm = TRUE),
            range = diff(range(percent.mt, na.rm = TRUE)),
            quantile = list(quantile(percent.mt, probs = c(0.25, 0.5, 0.75), na.rm = TRUE))) %>%
  unnest_wider(quantile)

meta <- seurat@meta.data
meta %>% summarize(mean = mean(percent.ribo, na.rm = TRUE),
            median = median(percent.ribo, na.rm = TRUE),
            min = min(percent.ribo, na.rm = TRUE),
            max = max(percent.ribo, na.rm = TRUE),
            range = diff(range(percent.ribo, na.rm = TRUE)),
            quantile = list(quantile(percent.ribo, probs = c(0.25, 0.5, 0.75), na.rm = TRUE))) %>%
  unnest_wider(quantile)

meta <- seurat@meta.data
meta %>% summarize(mean = mean(Complexity, na.rm = TRUE),
            median = median(Complexity, na.rm = TRUE),
            min = min(Complexity, na.rm = TRUE),
            max = max(Complexity, na.rm = TRUE),
            range = diff(range(Complexity, na.rm = TRUE)),
            quantile = list(quantile(Complexity, probs = c(0.25, 0.5, 0.75), na.rm = TRUE))) %>%
  unnest_wider(quantile)



###Filter data set, pick metrics you want to change and replot
seurat <- subset(seurat, subset = nFeature_RNA > 200 & nFeature_RNA < 10000 & percent.mt < 10 & Complexity > 0.8)
VlnPlot(seurat, features = c("nFeature_RNA", "percent.mt","percent.ribo", 'Complexity'), ncol = 4)
seurat
```

```
#SCT transformation and initial clustering


seurat <- SCTransform(seurat, vst.flavor = "v2", verbose = FALSE, variable.features.n = 2000) ###use SCT to normalize the data, with 2000 variable features.

seurat <- RunPCA(seurat, verbose = FALSE) ##Run PCA

DimHeatmap(seurat, dims = 1:11, cells = 500, balanced = TRUE, fast = FALSE)##Look at PVA with heatmap

ElbowPlot(seurat) ##look at elbow plot to see how many features to use

seurat <- FindNeighbors(seurat, dims = 1:13)

seurat <- FindClusters(seurat, resolution = c(0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35))

seurat <- RunUMAP(seurat, dims = 1:13)

seurat@meta.data

DimPlot(seurat, reduction = "umap", group.by = 'SCT_snn_res.0.15')


#DOUBLETFINDER


##detect doublets

# define the expected number of doublet cells.

nExp <- round(ncol(seurat) * 0.048)  # expect 4.8% doublets (https://kb.10xgenomics.com/hc/en-us/articles/360001378811-What-is-the-maximum-number-of-cells-that-can-be-profiled-)

seurat <- doubletFinder(seurat, PCs = 1:13, pN = 0.25, pK = 0.1, nExp = nExp, reuse.pANN = FALSE, sct = TRUE)


DF.name = colnames(seurat@meta.data)[grepl("DF.classification", colnames(seurat@meta.data))]

DimPlot(seurat, group.by = DF.name)

VlnPlot(seurat, features = "nFeature_RNA", group.by = DF.name, pt.size = 0.1)

head(seurat)


####Count how many singlets and how many doublets

plyr::count(seurat@meta.data$DF.classifications_0.25_0.1_216)
```

```r
##Keep only singlet

seurat = seurat[, seurat@meta.data[, DF.name] == "Singlet"]

dim(seurat)


# Re-cluster after doublet removal

##Run SCT transformation and identify clusters


seurat <- SCTransform(seurat, vst.flavor = "v2", verbose = FALSE, variable.features.n = 2000) ###use SCT to normalize the data, with 2000 variable features.

seurat <- RunPCA(seurat, verbose = FALSE) ##Run PCA

DimHeatmap(seurat, dims = 1:11, cells = 500, balanced = TRUE, fast = FALSE) ##Look at PVA with heatmap

ElbowPlot(seurat) ##look at elbow plot to see how many features to use

seurat <- FindNeighbors(seurat, dims = 1:13)

seurat <- FindClusters(seurat, resolution = c(0.02, 0.04, 0.06, 0.08, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35))

seurat<- RunUMAP(seurat, dims = 1:13)

seurat@meta.data

DimPlot(seurat, reduction = "umap", group.by = 'SCT_snn_res.0.15')


IMAT_JOVE<-c('DCN',

        'PDGFRA',

        'CD38','ATXN1', 'ZNF423',

        'PLIN1', 'ADIPOQ','LEP', 'SAA1',

        'PTPRC',

        'MRC1',

        'PECAM1','VWF',

        'PDGFRB', 'ACTA2',

        'MYH7B', 'LGR5',

        'MYLPF', 'MYH7'
```

```
)

# DotPlot

DefaultAssay(object = seurat) <- 'RNA'
seurat <- NormalizeData(seurat)
all.genes <- rownames(seurat)
seurat <- ScaleData(seurat, features = all.genes)

DotPlot(seurat, features = IMAT_JOVE, scale.min = 0, group.by = 'SCT_snn_res.0.15')  +  coord_flip() +
  scale_colour_gradientn(colours = rev(brewer.pal(n =11, name = 'PiYG')))

#Plot UMAP and DotPlot together

P1 <- DotPlot(seurat, features = IMAT_JOVE,
        scale.min = 0,
        group.by = "SCT_snn_res.0.15"
) +
  theme(axis.text.y.left = element_text(size=7), axis.title.y = element_blank())+
  coord_flip() +
  scale_colour_gradientn(colours = rev(brewer.pal(n =11, name = 'PRGn')))

P2 <- DimPlot(seurat, reduction = "umap", label = TRUE, group.by = "SCT_snn_res.0.15")

P2 + P1



##Identify what the different cell types are, this isn't necessary but is good practice to help
#you make sure you can identify the cells types before running ambient RNA control
seurat@meta.data <- seurat@meta.data %>% mutate(CellType = case_when(
  SCT_snn_res.0.15 == "0"  ~ "Pericyte/smooth muscle",
  SCT_snn_res.0.15 == "1"  ~ "Endothelial",
  SCT_snn_res.0.15 == "2"  ~ "Preadipocyte",
  SCT_snn_res.0.15 == "3"  ~ "Myonuclei",
  SCT_snn_res.0.15 == "4"  ~ "Stem/FAP",
  SCT_snn_res.0.15 == "5"  ~ "Muscle progenitor",
  SCT_snn_res.0.15 == "6"  ~ "Adipocyte_1",
  SCT_snn_res.0.15 == "7"  ~ "Immune",
  SCT_snn_res.0.15 == "8"  ~ "Adipocyte_2"

))


#AMBIENT RNA ADJUSTMENT

# Create a SingleCellExperiment object and run decontX
raw <-  Seurat::Read10X(data.dir = 'Raw')
dim(raw)

sce <- as.SingleCellExperiment(seurat)
sce1 <- SingleCellExperiment(list(counts = raw))

##Add celltypes from seurat to sce
colData(sce)$CellType <- seurat@meta.data$SCT_snn_res.0.15

##run decontx with the background droplets.
sce <- decontX(sce, z = colData(sce)$CellType, background = sce1)
colData(sce)$decontX_contamination

df <- as.data.frame(colData(sce))
```

```
ggplot(df, aes(x=decontX_contamination)) + geom_histogram(bins = 50, colour = 'black', fill = 'white')



###Convert SCE back to seurat if not using adjusted counts

seurat <- as.Seurat(sce)

VlnPlot(seurat, features = c("nFeature_RNA", "percent.mt", 'Complexity', 'decontX_contamination'), ncol = 4)



##Create a seurat object with the decontx counts and re run clustering and QC

seurat_d <- CreateSeuratObject(round(decontXcounts(sce)))


seurat_d[["percent.mt"]] <- PercentageFeatureSet(seurat_d, pattern = "^MT-")

RPS.genes <- grep(pattern = "^RPS", x = rownames(x = seurat_d), value = TRUE) ##Extract ribosomal RPS genes

RPL.genes <- grep(pattern = "^RPL", x = rownames(x = seurat_d), value = TRUE) ##Extract ribosomal RPL genes

ribo <- c(RPS.genes, RPL.genes) ##make vector of all ribosomal genes

seurat_d[["percent.ribo"]] <- PercentageFeatureSet(seurat_d, features = ribo) ##Add as percentage feature set


seurat_d@meta.data$Complexity <- log10(seurat_d@meta.data$nFeature_RNA) / log10(seurat_d@meta.data$nCount_RNA) ##Add complexity

head(seurat_d@meta.data)


##Plot metrics

VlnPlot(seurat_d, features = c("nFeature_RNA", "percent.mt", 'Complexity'), ncol = 3)

meta <- seurat_d@meta.data

meta %>% summarize(mean = mean(nFeature_RNA, na.rm = TRUE),

            median = median(nFeature_RNA, na.rm = TRUE),

            min = min(nFeature_RNA, na.rm = TRUE),

            max = max(nFeature_RNA, na.rm = TRUE),
            range = diff(range(nFeature_RNA, na.rm = TRUE)),
            quantile = list(quantile(nFeature_RNA, probs = c(0.25, 0.5, 0.75), na.rm = TRUE))) %>%
   unnest_wider(quantile)

meta <- seurat_d@meta.data
meta %>% summarize(mean = mean(nCount_RNA, na.rm = TRUE),
            median = median(nCount_RNA, na.rm = TRUE),
            min = min(nCount_RNA, na.rm = TRUE),
            max = max(nCount_RNA, na.rm = TRUE),
            range = diff(range(nCount_RNA, na.rm = TRUE)),
            quantile = list(quantile(nCount_RNA, probs = c(0.25, 0.5, 0.75), na.rm = TRUE))) %>%
   unnest_wider(quantile)

meta <- seurat_d@meta.data
meta %>% summarize(mean = mean(percent.mt, na.rm = TRUE),
            median = median(percent.mt, na.rm = TRUE),
            min = min(percent.mt, na.rm = TRUE),
            max = max(percent.mt, na.rm = TRUE),
            range = diff(range(percent.mt, na.rm = TRUE)),
            quantile = list(quantile(percent.mt, probs = c(0.25, 0.5, 0.75), na.rm = TRUE))) %>%
   unnest_wider(quantile)

meta <- seurat_d@meta.data
meta %>% summarize(mean = mean(percent.ribo, na.rm = TRUE),
            median = median(percent.ribo, na.rm = TRUE),
            min = min(percent.ribo, na.rm = TRUE),
            max = max(percent.ribo, na.rm = TRUE),
            range = diff(range(percent.ribo, na.rm = TRUE)),
            quantile = list(quantile(percent.ribo, probs = c(0.25, 0.5, 0.75), na.rm = TRUE))) %>%
   unnest_wider(quantile)
```

```
meta <- seurat_d@meta.data
meta %>% summarize(mean = mean(Complexity, na.rm = TRUE),
                   median = median(Complexity, na.rm = TRUE),
                   min = min(Complexity, na.rm = TRUE),
                   max = max(Complexity, na.rm = TRUE),
                   range = diff(range(Complexity, na.rm = TRUE)),
                   quantile = list(quantile(Complexity, probs = c(0.25, 0.5, 0.75), na.rm = TRUE))) %>%
  unnest_wider(quantile)

seurat_d

# Remove nuclei with less than 100 genes expressed
seurat_d <- subset(seurat_d, subset = nFeature_RNA > 100)

seurat_d

####Remove mitochondrial, hemoglobin and MALAT1/NEAT1 genes ####optional
remove_genes <- function(seurat){
  seurat <- seurat[!grepl("^MT-", rownames(seurat)), ] #filter mitochondrial genes
  seurat <- seurat[!grepl("^HB[^(P)]", rownames(seurat)), ]
  seurat <- seurat[!grepl("MALAT1", rownames(seurat)), ]
  seurat <- seurat[!grepl("NEAT1", rownames(seurat)), ]
  return(seurat)
}

seurat_d <- remove_genes(seurat_d)

seurat_d <- SCTransform(seurat_d, vst.flavor = "v2", verbose = FALSE, variable.features.n = 2000)
seurat_d <- RunPCA(seurat_d, verbose = FALSE)
```

```
DimHeatmap(seurat_d, dims = 1:15, cells = 500, balanced = TRUE, fast = FALSE)
ElbowPlot(seurat_d)
seurat_d <- FindNeighbors(seurat_d, dims = 1:13)
seurat_d <- FindClusters(seurat_d, resolution = c(0.04, 0.06, 0.08, 0.1, 0.15, 0.2, 0.25))
seurat_d <- RunUMAP(seurat_d, dims = 1:13)

DimPlot(seurat_d, reduction = "umap", group.by = 'SCT_snn_res.0.04')

DefaultAssay(object = seurat_d) <- 'RNA'
seurat_d <- NormalizeData(seurat_d)
all.genes <- rownames(seurat_d)
seurat_d <- ScaleData(seurat_d, features = all.genes)


DotPlot(seurat_d, features = IMAT, scale.min = 0, group.by = 'SCT_snn_res.0.04')  +  coord_flip() +
  scale_colour_gradientn(colours = rev(brewer.pal(n =11, name = 'PiYG')))


#Plot UMAP and DotPlot together

P1 <- DotPlot(seurat_d, features = IMAT_JOVE,
             scale.min = 0,
             group.by = "SCT_snn_res.0.04"
) +
  theme(axis.text.y.left = element_text(size=7), axis.title.y = element_blank())+
  coord_flip() +
  scale_colour_gradientn(colours = rev(brewer.pal(n =11, name = 'PRGn')))

P2 <- DimPlot(seurat_d, reduction = "umap", label = TRUE, group.by = "SCT_snn_res.0.04")
```

```
P2 + P1


##Identify what the different cell types are
seurat_d@meta.data <- seurat_d@meta.data %>% mutate(CellType = case_when(
  SCT_snn_res.0.04 == "0"  ~ "Endothelial",
  SCT_snn_res.0.04 == "1"  ~ "Pericyte/smooth muscle",
  SCT_snn_res.0.04 == "2"  ~ "Preadipocyte",
  SCT_snn_res.0.04 == "3"  ~ "Myonuclei",
  SCT_snn_res.0.04 == "4"  ~ "Stem/FAP",
  SCT_snn_res.0.04 == "5"  ~ "Muscle progenitor",
  SCT_snn_res.0.04 == "6"  ~ "Adipocyte_1",
  SCT_snn_res.0.04 == "7"  ~ "Immune",
  SCT_snn_res.0.04 == "8"  ~ "Adipocyte_2"
))


# reorder clusters
seurat_d@meta.data$CellType <- factor(seurat_d@meta.data$CellType,
                                      levels=c("Stem/FAP",
                                               "Preadipocyte",
                                               "Adipocyte_1",
                                               "Adipocyte_2",
                                               "Immune",
                                               "Endothelial",
                                               "Pericyte/smooth muscle",
                                               "Muscle progenitor",
                                               "Myonuclei"))




P1<- DotPlot(seurat_d, features = IMAT_JOVE, scale.min = 0, group.by = 'CellType')  +  coord_flip() +
  theme(text=element_text(size=10),
        axis.text.x = element_text(colour="black", size=11, angle = 45, hjust=1))+
  scale_colour_gradientn(colours = rev(brewer.pal(n =11, name = 'PiYG'))) +
  theme(axis.text.y.left = element_text(size=10), axis.title.y = element_blank())+
  coord_flip() +
  scale_colour_gradientn(colours = rev(brewer.pal(n =11, name = 'PRGn')))

P2<-DimPlot(seurat_d, reduction = "umap", label = TRUE, label.size = 4, repel = TRUE,group.by = "CellType")+ NoLegend()

P2 + P1
```