Fig. 7a and b

```
library(Seurat)
library(dplyr)
library(gdata)
library(tidyverse)
library(magrittr)
```

1. 10x C-sec dataset

1-1. Load the 10x C-sec dataset

```
CS <- Read10X("../Run2_CS/")
CS <- CreateSeuratObject(CS, min.cells= 3,  min.features = 200, project = "Run2_CS
")
CS
```

1-2. Pre-processing

```
# The percentage of reads that map to the mitochondrial genome
CS[["percent.mt"]] <- PercentageFeatureSet(object = CS, pattern = "^MT-")
# Filter cells that have unique feature counts over 7000 or less than 200, or >25%
mitochondrial counts
CS <- subset(x = CS, subset = nFeature_RNA > 200 & nFeature_RNA < 7000 & percent.m
t < 25)
```

1-3. Normalizing the data and identification of highly variable features

```
CS <- NormalizeData(object = CS, normalization.method = "LogNormalize", scale.fact
or = 10000)
CS <- FindVariableFeatures(object = CS, selection.method = "vst", nfeatures = 2000
)
```

2. Smart-seq2 FGC dataset

2-1. Load the Smart-seq2 FGC dataset

```
# UMI count and final cluster data were downloaded from http://github.com/zorrodon
g/germcell
Li.original<-read.delim("../190529Li-Run3/FGC_umi_counts.xls",row.names = 1)
Li.original <- as(as.matrix(Li.original), "dgCMatrix")
Li <- CreateSeuratObject(Li.original,project="Li",assay = "RNA")
Li

# Final cluster
final <- read.delim("../190529Li-Run3/Li-final-cluster.txt",header = 1,sep="\t",ch
eck.names = F)
```

2-2. Pre-processing

```
# The percentage of reads that map to the mitochondrial genome
Li[["percent.mt"]] <- PercentageFeatureSet(object = Li, pattern = "^MT-")
# but there is no mitochondrial genome here.

# Extract only female samples
Li.Female<-subset(Li, orig.ident=="F")
Li.Female
# 2629 -> 1435 samples.

Li.Female@meta.data<-merge(Li.Female@meta.data,final,by.x = 0,by.y = "sample",sort
= F,all=T) %>%
mutate(cluster = fct_explicit_na(cluster, na_level = "Female_NA"))%>% column_to_ro
wnames("Row.names")

VlnPlot(object = Li.Female, features = c("nFeature_RNA", "nCount_RNA"), ncol = 2)
FeatureScatter(object = Li.Female, feature1 = "nCount_RNA", feature2 = "nFeature_R
NA")

# Filter cells that have unique feature counts over 9000 or less than 2000, or ove
r 1100000 transcripts
Li.Female <- subset(x = Li.Female, subset = nFeature_RNA > 2000 & nFeature_RNA < 9
000
                    & nCount_RNA < 1100000)
```

2-3. Normalizing the data and identification of highly variable features

```
Li.Female <- NormalizeData(object = Li.Female, normalization.method = "LogNormaliz
e",
                           scale.factor = 10000)
Li.Female <- FindVariableFeatures(object = Li.Female, selection.method = "vst", nf
eatures = 2000)
```

3. Integration

3-1. Integration with FindIntegrationAnchors and IntegrateData function

```
Li.anchors <- FindIntegrationAnchors(object.list = c(CS,Li.Female), dims = 1:30)
Li.integrated <- IntegrateData(anchorset = Li.anchors, dims = 1:30)

Li.integrated@meta.data %<>% rownames_to_column("cell") %>%
mutate(cluster = fct_explicit_na(cluster, na_level = "Run2_CS")) %>% column_to_row
names("cell")
```

3-2. Cell cycle regression

```
s.genes <- cc.genes$s.genes
g2m.genes <- cc.genes$g2m.genes
Li.integrated<-CellCycleScoring(Li.integrated, s.features = s.genes, g2m.features
= g2m.genes,
                                set.ident = TRUE)
DefaultAssay(Li.integrated) <- "integrated"
Li.integrated <- ScaleData(Li.integrated, vars.to.regress = c("S.Score", "G2M.Scor
e"),
                           features = rownames(Li.integrated))
```

3-3. Standard workflow for visualization

```
Li.integrated <- RunPCA(Li.integrated, npcs = 30)
Li.integrated <- RunUMAP(Li.integrated, reduction = "pca", dims = 1:30)
```

4. Final visualization

4-1. Colored by original (adult or fetal)

```
svg("Fig7a.svg", width=10, height = 8)
DimPlot(Li.integrated, reduction = "umap", group.by = "orig.ident", pt.size =1)+
scale_color_manual(labels=c("Fetal cells", "Adult cells (C-sec)"), values = c("bla
ck","gray"))+
guides(colour = guide_legend(override.aes = list(size=8)))+theme(axis.text = eleme
nt_text(
size = 18, color="black"), axis.title = element_text(size = 24, color="black"),asp
ect.ratio=1,
legend.text = element_text(size = 22),legend.position = c(0.7, 0.12),legend.key.he
ight = unit(1, 'cm'))+
xlab("UMAP 1")+ylab("UMAP 2")
dev.off()
```

4-2. Colored by clusters

```
Li.integrated <- FindNeighbors(object = Li.integrated, dims = 1:30)
Li.integrated <- FindClusters(object = Li.integrated, resolution = 0.9)
DimPlot(Li.integrated, reduction = "umap", label = TRUE)
Li.integrated <- RenameIdents(object = Li.integrated ,'0' = 'stroma', '1' = 'strom
a', '2' = 'stroma',
'3' = 'stroma', '4' = 'stroma', '5' =  'stroma', '6' = 'perivascular', '7' = 'endo
thelial',
'8' = 'stroma', '9' = 'perivascular', '10' = 'mitotic FGCs', '11' = 'granulosa', '
12' = 'mitotic FGCs',
'13' = 'stroma','14' = 'meiotic FGCs', '15' = 'RA responsive FGCs', '16' = 'immune
','17' = 'oocytes' ,
'18' = 'immune')
```

```
svg("Fig7b.svg", width=10, height = 8)
DimPlot(Li.integrated, reduction = "umap", pt.size =1, label = FALSE, order = c('m
itotic FGCs',
'RA responsive FGCs', 'meiotic FGCs', 'oocytes', 'immune','granulosa','endothelial
','perivascular',
'stroma'), cols = c("grey", "#C49A00", "#53B400","#00B6EB", "#A58AFF","#FB61D7","#
FF9933","#CCCC00",
"#00FF80"))+ guides(colour = guide_legend(override.aes = list(size=8),reverse = TR
UE))+theme(
axis.text = element_text(size = 18,color="black"), axis.title = element_text(size
= 24, color="black"),
aspect.ratio=1,legend.text = element_text(size = 22),legend.key.height = unit(1, '
cm'))+
xlab("UMAP 1")+ylab("UMAP 2")
dev.off()
```