Pancreas islet integration

Weixu Wang

This tutorial is for users need to integrate more than three batches with more cell types. To deal with this, ICAnet need to perform preprocessing(denoise) on each datasets, and introduced singular vector decomposition to perform feature reductions. Here, I used three pancreas islet single cell datasets produced by different lab (named as baron et al, muraro et al, and segerostople et al) to illustrate how ICAnet perform batch effects correction on these dataset.

first, load the dataset, all these datasets can be download from
https://hemberg-lab.github.io/scRNA.seq.datasets/

```
baron <- readRDS("baron-human.rds")
muraro <- readRDS("muraro.rds")
segerstolpe <- readRDS("segerstolpe.rds")
baron_exp <- SingleCellExperiment::counts(baron)
segerstolpe_exp <- SingleCellExperiment::counts(segerstolpe)
muraro_exp <- normcounts(muraro)
geneID <- rownames(muraro_exp)
geneID_str <- strsplit(geneID,split="_")
for(i in 1:length(geneID)){
  geneID[i] <- geneID_str[[i]][1]
 }
 rownames(muraro_exp) <- geneID
geneID <- c(rownames(baron_exp),rownames(muraro_exp),rownames(segerstolpe_exp))
geneID <- names(table(geneID)[table(geneID)==3])
baron_exp <- baron_exp[geneID,]
segerstolpe_exp <- segerstolpe_exp[geneID,]
muraro_exp <- muraro_exp[geneID,]
```

Define batch vector and cell type vector

```
batch<-
c(baron$human,rep("segerstolpe",length(sample_infor$Characteristics.individual.))    ,
rep("muraro",length(muraro$donor)))
celltype <- c(as.character(as.matrix(baron$cell_type1)),
           as.character(as.matrix(segerstolpe$cell_type1)),
           as.character(as.matrix(muraro$cell_type1)))
pancreas <- CreateSeuratObject(cbind(baron_exp,segerstolpe_exp,muraro_exp))
pancreas$batch <- batch
pancreas$celltype <- celltype
```

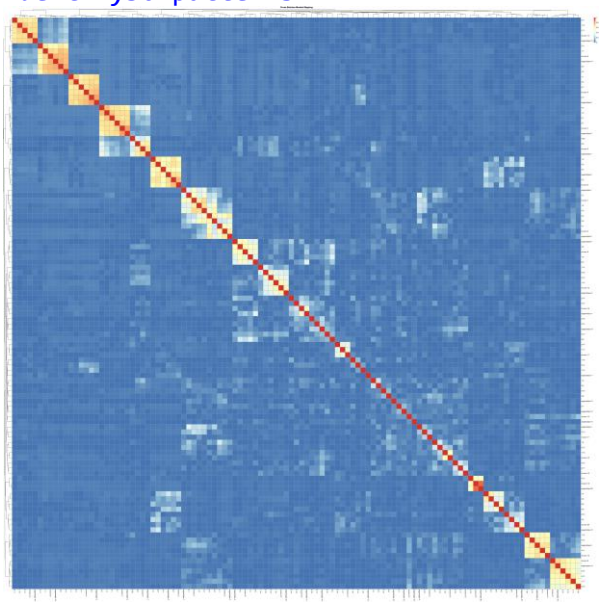
Running Seurat V3, perform scaling on each batch and using consensus highly variable genes to merge these datasets.

```
pancreas.list <- SplitObject(pancreas, split.by = "batch")
for (i in 1:length(pancreas.list)) {
    pancreas.list[[i]] <- NormalizeData(pancreas.list[[i]], verbose = FALSE)
    pancreas.list[[i]] <- FindVariableFeatures(pancreas.list[[i]], selection.method =
"vst",
        nfeatures = 8000, verbose = FALSE)
}
#####Integrate pancreas list
pancreas.all <- ScaleData(pancreas.list[[1]])
```

```
pancreas.all <- pancreas.all@assays$RNA@scale.data
for (i in 2:6) {
  pancreas.set <- ScaleData(pancreas.list[[i]])
  pancreas.set <- pancreas.set@assays$RNA@scale.data
  geneSet <- intersect(rownames(pancreas.all),rownames(pancreas.set))
  pancreas.all <- cbind(pancreas.all[geneSet,],pancreas.set[geneSet,])
}
pancreas.all <- CreateSeuratObject(pancreas.all)
pancreas.all$batch <- batch
pancreas.all$celltype <- celltype
```

Using ICAnet to learn expression programs on each dataset

```
ica.pancreas<-ICAcomputing2FastICA(pancreas.all,ICA.type="JADE",seurat.obj = TRUE,two.stage=FALSE)
ica.filter <- CrossBatchGrouping(ica.pancreas$ica.pooling,cor="spearman")
Identify30 patterns
```



We further select the clusters which it has more than one program. which represent these patterns are shared across different batches.

```
ica.filter<-
ica.filter$ica.filter[,as.numeric(names(table(ica.filter$cluster))[table(Ica.filter$cluster)>1])]
```

Running ICAnet

```
pancreas.all <- RunICAnet(pancreas.all,Ica.filter$ica.filter,species = 9606,W.top=2,aucMaxRank = 300)
```

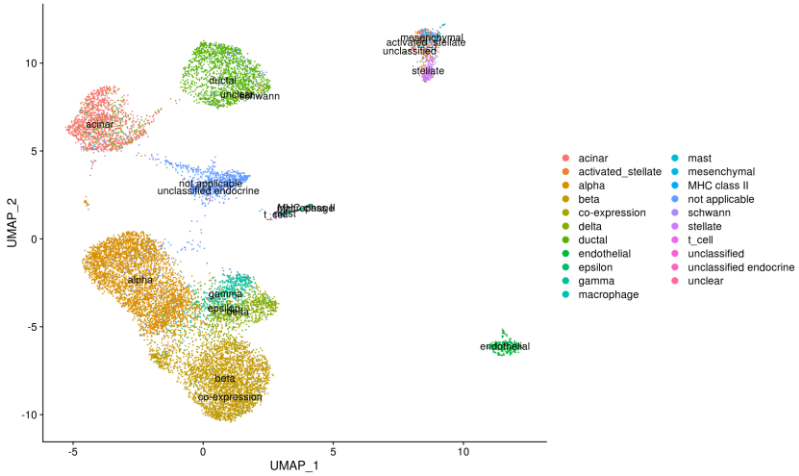
Check module dimensions

```
dim(pancreas.all)
```

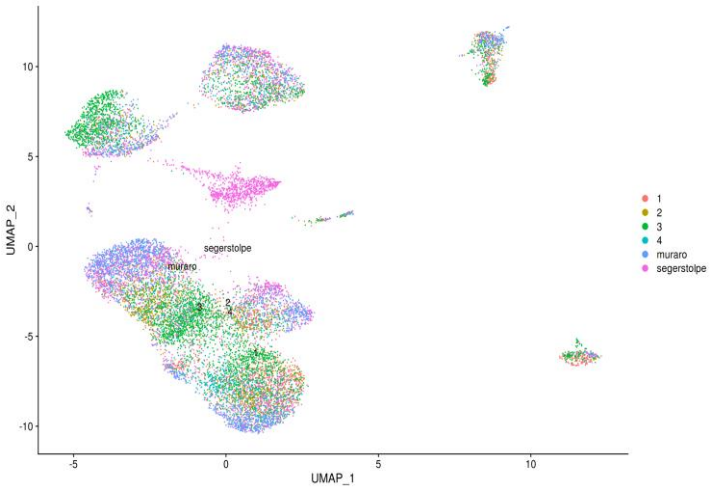
[1] 498 14209

ICAnet identify 498 module from the dataset, we further used module-cell matrix to perform SVD dimension reduction, ICAnet provide function RunModuleSVD to perform feature reductions, the parameter power is used to change the signal smoothness on specturm, if this parameter getting larger, ICAnet will enhance the signal of top singular vectors, here, I choose the power equal to 0.5

```
pancreas.all <- RunModuleSVD(save,nu=30,power=0.5)
pancreas.all <- RunUMAP(pancreas.all, dims = 1:20,reduction="SVD",reduction.name = "UMAP", reduction.key = "UMAP_")
```



As we can see the most cells are gouped according to the cell type, we can check their batch id to see if batch effects are corrected



We further perform clustering to show that ICAnet also benefits cell clustering, using lovain clustering algorthm

```
pancreas.all <- FindNeighbors(pancreas.all, dims = 1:20,reduction="SVD")
pancreas.all <- FindClusters(pancreas.all, resolution = 0.25,algorithm=2)
adjustedRandIndex(save@active.ident,as.numeric(as.factor(save$celltype)))
```

```
[1] 0.8028749
```

One thing need to be noted that, in this tutorial, we used consensus highly variable genes to denoise the dataset. User also can use RMT based denoise algorithm ‘randomly’ to denoise their dataset, which require gene count matrix, and saved as txt file .

```
source_python(denoise_read.py)
data <- denoise_read(readPath)
```