PDAC_Hwang2022_process

December 25, 2025

1 Of note

content in obj.PDAC@assays$RNA@counts: “UMI counts were normalized by the total number of UMIs per nucleus and converted to transcripts-per-10,000 (TP10K) as the final expression unit”

2 load data

[ ]:
```r
## gene info
feature <- read.delim('/project/sex_cancer/data/PDAC_Hwang2022/Group1/genes.
 ↪txt', header = F, row.names = NULL)

## obj1
exp <- Matrix::readMM('/project/sex_cancer/data/PDAC_Hwang2022/Group1/
 ↪Exp_data_TP10K_1.mtx')
meta <- read.csv('/project/sex_cancer/data/PDAC_Hwang2022/Group1/Cells1.csv')␣
 ↪%>% transform(barcode = cell_name) %>% column_to_rownames('cell_name')
colnames(exp) <- rownames(meta)
rownames(exp) <- feature$V1
obj1 <- CreateSeuratObject(counts = exp, meta.data = meta, project = "PDAC")

## obj2
exp <- Matrix::readMM('/project/sex_cancer/data/PDAC_Hwang2022/Group2/
 ↪Exp_data_TP10K_2.mtx')
meta <- read.csv('/project/sex_cancer/data/PDAC_Hwang2022/Group2/Cells2.csv')␣
 ↪%>% transform(barcode = cell_name) %>% column_to_rownames('cell_name')
colnames(exp) <- rownames(meta)
rownames(exp) <- feature$V1
obj2 <- CreateSeuratObject(counts = exp, meta.data = meta, project = "PDAC")

## obj2
exp <- Matrix::readMM('/project/sex_cancer/data/PDAC_Hwang2022/Group3/
 ↪Exp_data_TP10K_3.mtx')
meta <- read.csv('/project/sex_cancer/data/PDAC_Hwang2022/Group3/Cells3.csv')␣
 ↪%>% transform(barcode = cell_name) %>% column_to_rownames('cell_name')
colnames(exp) <- rownames(meta)
rownames(exp) <- feature$V1
obj3 <- CreateSeuratObject(counts = exp, meta.data = meta, project = "PDAC")
```

```
## all merge into 1
obj.PDAC <- merge(obj1, c(obj2, obj3))
```

[ ]:
```
info <- read_xlsx('/project/sex_cancer/data/PDAC_Hwang2022/
 ↪PDAC_WilliamL2022_PatientInfo.xlsx',skip = 1)
info1 <- info[grepl('PDAC_',info$ID),]

meta <- obj.PDAC@meta.data %>% transform(barcode2 = rownames(.))
sample=sapply(info1$ID,function(x){
                    parts=strsplit(x,split = '_')[[1]]
                    new=paste(parts[2:3],collapse = '')
                    return(new)
})
info2 <- cbind(sample,info1) %>%
         .[which(.$sample %in% meta$sample),]
obj.PDAC@meta.data <- merge(meta, info2, by = 'sample',all = TRUE) %>%␣
 ↪column_to_rownames('barcode2') %>% .[colnames(obj.PDAC),]
```

3 modify meta.data

[ ]:
```
## remove useless meta.data
obj.PDAC@meta.data <- obj.PDAC@meta.data %>%
                      dplyr::select(-c('orig.ident', 'complexity', 'umap1',␣
 ↪'umap2', 'g1s_score', 'g2m_score', 'cell_cycle_phase', 'nFeature_RNA',
                                       'mp_top', 'mp_top_score',␣
 ↪'mp_assignment', 'mCT', 'disease'))
```

[ ]:
```
obj.PDAC@meta.data  <-  obj.PDAC@meta.data %>%
                        dplyr::rename(c('Chemistry' = '10x Chemistry',␣
 ↪'SampleID' = 'sample', 'Disease' = 'source')) %>%
                        mutate(Chemistry = case_when(Chemistry == 'v2' ~ "10x␣
 ↪3' v2",
                                                     Chemistry == 'v3' ~ "10x␣
 ↪3' v3",
                                                     TRUE ~ 'Others')) %>%
                        mutate(Sex = case_when(Sex == 'Female' ~ 'F', Sex ==␣
 ↪'Male' ~ 'M', TRUE ~ 'Others')) %>%
                        transform(Cohort = 'PDAC_Hwang2022')
```

4 filter sample

[ ]:
```
obj.PDAC <- obj.PDAC %>% subset(Neoadjuvant == 'None')
obj.PDAC@meta.data <- obj.PDAC@meta.data %>% transform(SampleType = 'tumor')
```

5 UMAP visualization

[ ]:
```
obj.h5ad <- anndata::read_h5ad('/project/sex_cancer/data/PDAC_Hwang2022/
↪GSE202051_totaldata-final-toshare.h5ad')
umap_emb <- obj.h5ad$obsm$X_umap %>% as.data.frame() %>%
            `rownames<-`(rownames(obj.h5ad$X)) %>% `colnames<-`(c('umap_1',␣
↪'umap_2'))
umap_emb <- umap_emb %>% .[colnames(obj.PDAC),] %>% as.matrix()
## add UMAP embeddings
Idents(obj.PDAC) <- obj.PDAC$barcode
obj.PDAC[['umap']] <- CreateDimReducObject(embeddings = umap_emb[colnames(obj.
↪PDAC),] , key = 'umap_', assay = 'RNA')
```

6 cell type annotation

6.1 assign oCT

[ ]:
```
obj.PDAC@meta.data <- obj.PDAC@meta.data %>%
                      mutate(oCT = cell_subtype)  %>%
                      mutate(dCT = case_when(cell_subtype %in% c('Ductal␣
↪(atypical)', 'Ductal', 'Acinar', 'ADM') ~ 'Epi',
                                             cell_subtype %in% c('Malignant') ~␣
↪'Epi',
                                             cell_subtype %in% c('CD8+ T') ~␣
↪'CD8T',
                                             cell_subtype %in% c('CD4+ T') ~␣
↪'CD4T',
                                             cell_subtype %in% c('Treg') ~␣
↪'Treg',
                                             cell_subtype %in% c('NK_cell') ~␣
↪'NK',
                                             cell_subtype %in% c('B_cell') ~␣
↪'B',
                                             cell_subtype %in% c('Plasma') ~␣
↪'Plasma',

                                             cell_subtype %in% c('Macrophage')␣
↪~ 'Mph',
                                             cell_subtype %in% c('Dendritic') ~␣
↪'DC',
```

```
                                                cell_subtype %in% c('Neutrophil')␣
↪~ 'Neu',
                                                cell_subtype %in% c('Mast') ~␣
↪'Mast',

                                                cell_subtype %in% c('myCAF',␣
↪'CAF') ~ 'Fibro',
                                                cell_subtype %in% c('Pericyte') ~␣
↪'Pericyte',
                                                cell_subtype %in% c('Vascular',␣
↪'Lymphatic') ~ 'Endo',
                                                cell_subtype %in% c('Schwann') ~␣
↪'Schwann',
                                                cell_subtype %in% c('Alpha',␣
↪'Beta', 'Delta', 'Epsilon', 'Gamma', 'Hormone-negative neuroendocrine',␣
↪'Intra-pancreatic neurons') ~ 'Neuron',
                                                cell_subtype %in% c('Adipocyte') ~␣
↪'Adipocyte',
                                                cell_subtype %in% c('Vascular␣
↪smooth muscle') ~ 'VSMC',
                                                TRUE ~ 'Others'))
```

6.2 check annotation

check cell type annotation provided in the original research via COSG

[ ]:
```
## check marker expression
marker_annotation <- readRDS("marker_annotation.rds")

obj <- obj.PDAC
DefaultAssay(obj) <- "RNA"
obj <- obj %>% NormalizeData(normalization.method = "LogNormalize", scale.
 ↪factor = 10000, verbose = F)
Idents(obj) <- ext_list(obj$oCT)

marker_oCT <- obj %>%
              cosg(groups = "all", assay = "RNA", slot = "data",
              mu = 10, ## The penalty factor to penalize gene expression in␣
 ↪cells not belonging to the cluster of interest
              n_genes_user = 50, # Number of top ranked genes returned in the␣
 ↪result
              remove_lowly_expressed=T, # If TRUE, genes that express a␣
 ↪percentage of target cells smaller than a specific value (expressed_pct) are␣
 ↪not considered as marker genes for the target cells. The default value is␣
 ↪TRUE.
```

```
            expressed_pct=0.1) # If TRUE, genes that express a percentage of␣
 ↪target cells smaller than a specific value (expressed_pct) are not␣
 ↪considered as marker genes for the target cells.
marker_oCT <- cbind(marker_oCT[[1]] %>% melt(id.vars = NULL) %>% dplyr::
 ↪rename(c("oCT" = "variable", "marker" = "value")),
                    marker_oCT[[2]] %>% melt(id.vars = NULL) %>% dplyr::
 ↪select(-"variable") %>% dplyr::rename(c("COSGscore" = "value"))) %>%
             mutate(Cohort = unique(obj$Cohort)) %>% mutate(oCT =␣
 ↪ext_list(oCT))

oCT_marker <- marker_oCT
oCT_list <- unique(oCT_marker$oCT)
lapply(oCT_list, function(x){
        check <- oCT_marker %>% subset(oCT == x & marker %in%␣
 ↪marker_annotation[[x]])
        ifelse(nrow(check) == 0, print(x), return(check))
})
```

6.3 assign mCT

[ ]:
```
obj.PDAC@meta.data <- obj.PDAC@meta.data %>%
                      mutate(mCT = case_when(cell_subtype %in% c('Ductal␣
 ↪(atypical)', 'Ductal', 'Acinar', 'ADM') ~ 'Epi',
                                             cell_subtype %in% c('Malignant') ~␣
 ↪'Epi',

                                             cell_subtype %in% c('CD8+ T') ~␣
 ↪'CD8T',
                                             cell_subtype %in% c('CD4+ T') ~␣
 ↪'CD4T',
                                             cell_subtype %in% c('Treg') ~␣
 ↪'Treg',
                                             cell_subtype %in% c('NK_cell') ~␣
 ↪'NK',
                                             cell_subtype %in% c('B_cell',␣
 ↪'Plasma') ~ 'B',

                                             cell_subtype %in% c('Macrophage')␣
 ↪~ 'Mph',
                                             cell_subtype %in% c('Dendritic') ~␣
 ↪'DC',
                                             cell_subtype %in% c('Neutrophil')␣
 ↪~ 'Neu',
                                             cell_subtype %in% c('Mast') ~␣
 ↪'Mast',
```

```
                                                cell_subtype %in% c('myCAF',␣
↪'CAF') ~ 'Fibro',
                                                cell_subtype %in% c('Pericyte') ~␣
↪'Pericyte',
                                                cell_subtype %in% c('Vascular',␣
↪'Lymphatic') ~ 'Endo',
                                                cell_subtype %in% c('Schwann') ~␣
↪'Schwann',
                                                cell_subtype %in% c('Alpha',␣
↪'Beta', 'Delta', 'Epsilon', 'Gamma', 'Hormone-negative neuroendocrine',␣
↪'Intra-pancreatic neurons') ~ 'Neuron',
                                                cell_subtype %in% c('Adipocyte') ~␣
↪'Adipocyte',
                                                cell_subtype %in% c('Vascular␣
↪smooth muscle') ~ 'SMC',
                                                TRUE ~ 'Others'))
```

6.4 assign gCT

[ ]:
```
obj.PDAC@meta.data <- obj.PDAC@meta.data %>%
                      mutate(gCT = case_when(mCT %in% c('Epi') ~ 'Tumor',
                                                mCT %in% c('CD8T', 'CD4T', 'Treg',␣
↪'NK', 'B', 'Mph', 'DC', 'Neu', 'Mast') ~ 'Immune',
                                                mCT %in% c('Fibro', 'Pericyte',␣
↪'Endo', 'Schwann', 'Neuron', 'Adipocyte', 'SMC') ~ 'Stromal',
                                                TRUE ~ 'Others'))
```

7 save

[ ]:
```
saveRDS(obj.PDAC, 'obj.PDAC.use.rds')
```