GC_Kumar2022_process

December 25, 2025

1 load data

```
[ ]: obj.GC <- load3CAdata("/project/sex_cancer/data/GC_Kumar2022/")
     obj.GC
```

2 modify meta.data

```
[ ]: obj.GC@meta.data <- obj.GC@meta.data %>%
                         transform(orig.ident = sample) %>%
                         dplyr::rename(c('SampleID' = 'sample', 'DonorID' =␣
      ↪'patient')) %>%
                         transform(Cohort = 'GC_Kumar2022', Chemistry = "10x 5'")
```

```
[ ]: # add sample info
     sample_info <- read.csv('/project/sex_cancer/data/GC_Kumar2022/Samples.csv') %>%
                    dplyr::select(-c('n_cells', 'technology', 'cancer_type')) %>%
                    dplyr::rename(c('SampleID' = 'sample', 'SampleType' = 'source',␣
      ↪'DonorID' = 'patient',
                                    'Age' = 'age', 'Sex' = 'gender', 'Site' =␣
      ↪'site', 'MMR' = 'mmr', 'EBV' = 'ebv', 'IM_status' = 'im_status',
                                    'StageI' = 'stage', 'Laurens' = 'laurens')) %>%
                    mutate(SampleType = case_when(SampleType == "Primary_Tumor" ~␣
      ↪"tumor",
                                                  SampleType == "Primary_Normal" ~␣
      ↪"normal_adjacent",
                                                  SampleType == "Peritoneal_Tumor" ~␣
      ↪"tumor_peritoneal",
                                                  SampleType == "Peritoneal_Normal"␣
      ↪~ "normal_peritoneal",
                                                  TRUE ~ NA))

     obj.GC@meta.data <- obj.GC@meta.data %>%
                         transform(barcode2 = barcode) %>%
                         merge(., sample_info, by = c('SampleID', 'DonorID'), all =␣
      ↪TRUE) %>%
                         column_to_rownames('barcode2')
```

```
obj.GC@meta.data <- obj.GC@meta.data[colnames(obj.GC),]
```

3 filter sample

```
[ ]: obj.GC <- obj.GC %>% subset(SampleType %in% c('tumor_peritoneal',␣
      ↪'normal_peritoneal') == FALSE) ## & Laurens != 'Metastatic'
     obj.GC
```

```
[ ]: obj.GC@meta.data[,c('SampleID', 'StageI', 'Laurens', 'IM_status', 'Sex',␣
      ↪'Site', 'SampleType')] %>% .[!duplicated(.$SampleID),]  %$%
     table(.$SampleType, .$Sex)
     obj.GC@meta.data[,c('SampleID', 'StageI', 'Laurens', 'IM_status', 'Sex',␣
      ↪'Site', 'SampleType')] %$% table(.$SampleType)
```

4 cell type annotation

4.1 assign oCT

```
[ ]: obj.GC@meta.data <- obj.GC@meta.data %>%
                         mutate(oCT = ifelse(cell_cluster == '', 'Unknown',␣
      ↪cell_cluster)) %>%
                         mutate(dCT = case_when(cell_type %in% c('Epithelial') ~␣
      ↪'Epi',
                                                   cell_cluster %in% c('LB') ~ 'B',
                                                   cell_cluster %in% c('PL1', 'PL2',␣
      ↪'PL3', 'PL4', 'PL5') ~ 'Plasma',
                                                   cell_cluster %in% c('LNK1', 'LNK2',␣
      ↪'LNK3', 'LNK4') ~ 'NK',
                                                   cell_cluster %in% c('LT1', 'LT2') ~␣
      ↪'CD8T',
                                                   cell_cluster %in% c('LT3', 'LT4') ~␣
      ↪'CD4T',
                                                   cell_cluster %in% c('LT5') ~ 'Treg',
                                                   cell_cluster %in% c('LPT') ~␣
      ↪'T_proliferation',
                                                   cell_cluster %in% c('MYD') ~ 'DC',
                                                   cell_cluster %in% c('MYM1', 'MYM2',␣
      ↪'MYM3') ~ 'Mph',
                                                   cell_cluster %in% c('MYMAST') ~␣
      ↪'Mast',
                                                   cell_cluster %in% c('STE1', 'STE2')␣
      ↪~ 'Endo',
                                                   cell_cluster %in% c('STF1', 'STF2',␣
      ↪'STF3', 'STF4') ~ 'Fibro',
                                                   TRUE ~ 'Others'
                                                   ))
   jyyht
   table(obj.GC$cell_cluster, obj.GC$dCT)
```

4.2 check annotation

check cell type annotation provided in the original research via COSG

[ ]:

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

obj <- obj.GC
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))
})
```

4.3 assign mCT

[ ]:

```
obj.GC@meta.data <- obj.GC@meta.data %>%
                    mutate(mCT = case_when(dCT %in% c('B', 'Plasma') ~ 'B',
                                           TRUE ~ dCT))
table(obj.GC$dCT, obj.GC$mCT)
```

4.4 assign gCT

[ ]:

```
obj.GC@meta.data <- obj.GC@meta.data %>%
                    mutate(gCT = case_when(mCT %in% c('Epi') ~ 'Tumor',
                                           mCT %in% c('NK', 'B', 'Mph', 'CD8T', 'CD4T', 'Mast', 'DC', 'Treg', 'T_proliferation') ~ 'Immune',
                                           mCT %in% c('Endo', 'Fibro') ~ 'Stromal',
                                           TRUE ~ 'Others'
                                           ))
table(obj.GC$dCT, obj.GC$mCT)
```

4.5 discard unannotated cells

[ ]:

```
obj.GC <- obj.GC %>% subset(mCT != "Others")
table(obj.GC$mCT, useNA = "ifany")
```

5 UMAP visualization

[ ]:

```
obj.GC <- obj.GC %>%
          NormalizeData(normalization.method = "LogNormalize", scale.factor = 10000, verbose = F) %>%
          FindVariableFeatures(selection.method = "vst", nfeatures = 3000, verbose = F) %>%
          ScaleData(vars.to.regress = c("nCount_RNA"), verbose = F) %>%
          RunPCA(verbose = F) %>%
          RunHarmony(group.by.vars = "SampleID", plot_convergence = TRUE)

nPC <- min(PC_selection_harmony(obj.GC)$PCselect)
obj.GC <- obj.GC %>%
          RunUMAP(reduction = "harmony", dims = 1:nPC, umap.method = "uwot") %>%
          RunTSNE(reduction = "harmony", dims = 1:nPC)
colnames(obj.GC@meta.data)
```

[ ]:

```
options(repr.plot.height = 4, repr.plot.width = 30)
select <- 'umap'
DimPlot_scCustom(obj.GC, reduction = select, pt.size = 1, group.by = "gCT", label = F, label.size = 8, colors_use = pal_igv("default")(51))|
DimPlot_scCustom(obj.GC, reduction = select, pt.size = 1, group.by = "mCT", label = F, label.size = 8, colors_use = pal_igv("default")(51))|
DimPlot_scCustom(obj.GC, reduction = select, pt.size = 1, group.by = "SampleType", label = F, label.size = 8, colors_use = pal_igv("default")(51))|
DimPlot_scCustom(obj.GC, reduction = select, pt.size = 1, group.by = "SampleID", label = F, label.size = 8, colors_use = pal_igv("default")(51))
```

6 save

[ ]:

```
saveRDS(obj.GC, 'obj.GC.use.rds')
```