CRC_Pelka2021_process

December 25, 2025

1 load data

[ ]:
```
data1 <- load3CAdata('/project/sex_cancer/data/CRC_Pelka2021/Group1')
data2 <- load3CAdata('/project/sex_cancer/data/CRC_Pelka2021/Group2')
data3 <- load3CAdata('/project/sex_cancer/data/CRC_Pelka2021/Group3')
data4 <- load3CAdata('/project/sex_cancer/data/CRC_Pelka2021/Group4')
obj.CRC <- merge(data1, c(data2, data3, data4)) %>% UpdateSeuratObject()
```

[ ]:
```
## add sample info
sample_info <- read.csv('/project/sex_cancer/data/CRC_Pelka2021/
 ↪GSE178341_crc10x_full_c295v4_submit_metatables.csv')
```

[ ]:
```
## add cell type annotation
cell_anno <- read.csv('/project/sex_cancer/data/CRC_Pelka2021/
 ↪crc10x_full_c295v4_submit_cluster.csv')
nrow(cell_anno)
cell_anno %>% head(n = 2)
```

2 modify meta.data

[ ]:
```
meta_data <- cbind(sample_info %>% transform(barcode = cellID) %>%␣
 ↪column_to_rownames('cellID'),
                    cell_anno %>% column_to_rownames('sampleID')) %>%
              dplyr::rename(c('SampleID' = 'PatientTypeID', 'DonorID' = 'PID'))␣
 ↪%>%
              transform(SampleType = ifelse(SPECIMEN_TYPE == 'T', 'tumor',␣
 ↪'normal_adjacent')) %>%
              transform(Cohort = 'CRC_Pelka2021') %>%
              mutate(Chemistry = case_when(SINGLECELL_TYPE == 'SC3Pv2' ~ "10x 3'␣
 ↪v2",
                                           SINGLECELL_TYPE == 'SC3Pv3' ~ "10x 3'␣
 ↪v3",
                                           TRUE ~ 'unknown')) %>%
              .[colnames(obj.CRC),]
obj.CRC@meta.data <- meta_data
```

[ ]:
```
obj.CRC@meta.data[,c('SampleType', 'SampleID', 'Sex', 'MetastasisStatus')] %>% .
 ↪[!duplicated(.$SampleID),] %$% table(.$Sex, .$SampleType, useNA = 'ifany')
obj.CRC@meta.data %$% table(.$SampleType)
```

3 cell type annotation

3.1 assign oCT

[ ]:
```
obj.CRC@meta.data <- obj.CRC@meta.data %>%
                     dplyr::rename(c('oCT' = 'clMidwayPr', 'dCT' =␣
 ↪'cl295v11SubFull'))
```

3.2 check annotation

check cell type annotation provided in the original research via COSG

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

obj <- obj.CRC
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))
})
```

3.3 assign mCT

[ ]:
```
obj.CRC@meta.data <- obj.CRC@meta.data %>%
                     mutate(mCT = case_when(oCT %in% c('Epi') ~ 'Epi',
                                            oCT %in% c('TCD8') ~ 'CD8T',
                                            (oCT == 'TCD4' & dCT %in% c("cTNI08␣
 ↪(CD4+ Treg)", "cTNI09 (CD4+ Treg prolif)") == FALSE) ~ 'CD4T',
                                            dCT %in% c("cTNI08 (CD4+ Treg)",␣
 ↪"cTNI09 (CD4+ Treg prolif)") ~ 'Treg',
                                            oCT %in% c('TZBTB16') ~ 'T_ZBTB16',
                                            oCT %in% c('Tgd') ~ ' T',
                                            oCT %in% c('Macro') ~ 'Mph',
                                            oCT %in% c('Granulo') ~ 'Neu',
                                            oCT %in% c('SmoothMuscle') ~ 'SMC',
                                            oCT %in% c('Peri') ~ 'Pericyte',
                                            oCT %in% c('Plasma', 'B') ~ 'B',
                                            TRUE ~ oCT
                                            ))
unique(obj.CRC$mCT)
```

3.4 assign gCT

[ ]:
```
obj.CRC@meta.data <- obj.CRC@meta.data %>%
                     mutate(gCT = case_when(mCT %in% c('Epi') ~ 'Tumor',
                                            mCT %in% c('CD4T', 'Mph', 'Treg',␣
 ↪'DC', 'NK', 'Mono', 'CD8T', 'B', 'Mast', 'T_ZBTB16', ' T', 'Neu', 'ILC') ~␣
 ↪'Immune',
                                            mCT %in% c('Pericyte', 'Endo',␣
 ↪'Fibro', 'SMC', 'Schwann') ~ 'Stromal',
                                            TRUE ~ 'Others'
                                            ))
table(obj.CRC$mCT, obj.CRC$gCT, useNA = 'ifany')
```

4 UMAP visualization

[ ]:
```
# add nCount
nCount_RNA <- colSums(x = obj.CRC, slot = "counts")
obj.CRC <- obj.CRC %>% AddMetaData(metadata = nCount_RNA, col.name =␣
↪'nCount_RNA')
obj.CRC@meta.data %>% head(n = 2)
```

[ ]:
```
# run harmony
obj.CRC <- obj.CRC %>%
           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)
```

[ ]:
```
# run UMAP
nPC <- min(PC_selection_harmony(obj.CRC)$PCselect)
obj.CRC <- obj.CRC %>%
           RunUMAP(reduction = "harmony", dims = 1:nPC, umap.method = "uwot")␣
↪%>%
           RunTSNE(reduction = "harmony", dims = 1:nPC)
```

[ ]:
```
options(repr.plot.height = 5, repr.plot.width = 25)
DimPlot_scCustom(obj.CRC, pt.size = 1, group.by = "gCT", label = TRUE, label.
↪size = 4, colors_use = pal_igv("default")(51))|
DimPlot_scCustom(obj.CRC, pt.size = 1, group.by = "mCT", label = TRUE, label.
↪size = 4, colors_use = pal_igv("default")(51))|
DimPlot_scCustom(obj.CRC, pt.size = 1, group.by = "oCT", label = TRUE, label.
↪size = 4, colors_use = paletteer::paletteer_d('palettesForR::Web', 100))
```

5 save

[ ]:
```
saveRDS(obj.CRC, "obj.CRC.use.rds")
```