HNSCC_Choi2023_process

December 25, 2025

1 load data

[ ]:
```r
meta <- read.table('/project/sex_cancer/data/HNSCC_Choi2023/
 ↪GSE181919_Barcode_metadata.txt',header = T,sep = '\t') %>%
       rownames_to_column(meta,'barcode')
info <- read_xlsx('/project/sex_cancer/data/HNSCC_Choi2023/
 ↪HNSCC_Choi2023_PatientInfo.xlsx')
colnames(info)[1]='patient.id'
meta <- merge(meta,info,by = 'patient.id',all = TRUE)

counts <- read.table('/project/sex_cancer/data/HNSCC_Choi2023/
 ↪GSE181919_UMI_counts.txt',header=T,sep = '\t')
colnames(counts) <- gsub('\\.','-',colnames(counts))

obj.HNSCC <- CreateSeuratObject(counts = counts,meta.data = meta, min.cells =␣
 ↪0,min.features = 0)

## de-factor
obj.HNSCC@meta.data <- obj.HNSCC@meta.data %>% mutate_if(~!is.numeric(.),␣
 ↪ext_list)
```

2 filter sample

[ ]:
```r
obj.HNSCC <- obj.HNSCC %>% subset(tissue_type == 'CA' & Tumor_subsite == 'OC')␣
 ↪## remove non-primary samples
obj.HNSCC
```

[ ]:
```r
obj.HNSCC %$% table(.$patient_id, .$Sex) %>% as.data.frame() %>% subset(Freq>0)␣
 ↪%$% table(.$Var2)
```

3 modify meta.data

[ ]:
```
obj.HNSCC@meta.data <-  obj.HNSCC@meta.data %>%
                        dplyr::rename(c('SampleID' = 'sample_id', 'DonorID' =␣
 ↪'patient_id')) %>%
                        transform(Cohort = 'HNSCC_Choi2023', 'SampleType' =␣
 ↪'tumor', Tumor_subsite = 'Oral_cavity')
obj.HNSCC@meta.data %>% head(n = 2)
```

4 cell type annotation

4.1 assign oCT

[ ]:
```
obj.HNSCC@meta.data <-  obj.HNSCC@meta.data %>%
                        mutate(oCT = cell.type) %>%
                        mutate(dCT = case_when(oCT %in% c('T.cells') ~ 'T/NK',
                                               oCT %in% c('Malignant.cells') ~␣
 ↪'Epi',
                                               oCT %in% c('B_Plasma.cells') ~␣
 ↪'B/Plasma',
                                               oCT %in% c('Endothelial.cells')␣
 ↪~ 'Endo',
                                               oCT %in% c('Fibroblasts') ~␣
 ↪'Fibro',
                                               oCT %in% c('Macrophages') ~␣
 ↪'Mph',
                                               oCT %in% c('Dendritic.cells') ~␣
 ↪'DC',
                                               oCT %in% c('Mast.cells') ~␣
 ↪'Mast',
                                               oCT %in% c('Myocytes') ~␣
 ↪'Myocyte',
                                               TRUE ~ 'Others'))
table(obj.HNSCC$oCT, obj.HNSCC$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.HNSCC
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 NK/T

[ ]:
```
obj.HNSCC.nkt <- obj.HNSCC %>% subset(dCT == 'T/NK')
obj.HNSCC.nkt

obj.HNSCC.nkt <- obj.HNSCC.nkt %>%
                 NormalizeData(normalization.method = "LogNormalize", scale.
 ↪factor = 10000, verbose = F) %>%
                 FindVariableFeatures(selection.method = "vst", nfeatures =␣
 ↪1000, verbose = F) %>%
                 ScaleData(vars.to.regress = c("nCount_RNA"), verbose = F) %>%
                 RunPCA(verbose = F) %>%
                 RunHarmony(group.by.vars = "SampleID", plot_convergence = TRUE)
## cluster
nPC <- min(PC_selection_harmony(obj.HNSCC.nkt)$PCselect)
obj.HNSCC.nkt <- obj.HNSCC.nkt %>%
```

```
                  RunUMAP(reduction = "harmony", dims = 1:nPC, umap.method =␣
 ↪"uwot") %>%
                  RunTSNE(reduction = "harmony", dims = 1:nPC) %>%
                  FindNeighbors(reduction = "harmony", dims = 1:nPC) %>%
                  FindClusters(resolution=0.1) %>% FindClusters(resolution=0.2)␣
 ↪%>% FindClusters(resolution=0.3)
colnames(obj.HNSCC.nkt@meta.data) <- gsub("RNA_snn_res.0.","r0", colnames(obj.
 ↪HNSCC.nkt@meta.data))
colnames(obj.HNSCC.nkt@meta.data)
```

[ ]:
```
## marker expression
marker_list <- c('CD3D', 'CD3E', 'CD3G', 'TRDC',
                 'CD4','FOXP3', 'CTLA4',
                 'CD8A', 'CD8B', 'CD28', 'GZMA', 'GZMH',
                 'TIGIT', 'PDCD1', 'TCF7',
                 'GNLY', 'NKG7', 'KLRD1', 'NCAM1', 'FCGR3A', 'PRF1',
                 'MKI67', 'TOP2A', 'STMN1', 'TOX')

options(repr.plot.height = 8, repr.plot.width = 5)
VlnPlot(obj.HNSCC.nkt, group.by = 'r02', features = marker_list,
        pt.size = 0, cols = pal_igv("default")(51),
        slot = 'data',assay = 'RNA', raster=FALSE, stack = TRUE, flip = TRUE)+
theme(legend.position = 'none')
```

[ ]:
```
Idents(obj.HNSCC.nkt) <- factor(obj.HNSCC.nkt$r02, levels = 8:0)
cluster_deg <- FindAllMarkers(obj.HNSCC.nkt, assay = "RNA", slot = "data",
                              logfc.threshold = 0.25, min.pct = 0.1, test.use =␣
 ↪"wilcox")
rownames(cluster_deg) <- NULL
```

[ ]:
```
obj.HNSCC.nkt@meta.data <- obj.HNSCC.nkt@meta.data %>%
                           mutate(dCT = case_when(r02 %in% c('0', '3', '4',␣
 ↪'6') ~ 'CD8T',
                                                  r02 %in% c('1') ~ 'Treg',
                                                  r02 %in% c('2') ~ 'T_cycling',
                                                  r02 %in% c('5') ~ ' T',
                                                  r02 %in% c('7', '8') ~␣
 ↪'Others',
                                                  TRUE ~ 'Others'))
obj.HNSCC.nkt@meta.data <- obj.HNSCC.nkt@meta.data %>% dplyr::select(-c('r01',␣
 ↪'r02', 'r03', 'seurat_clusters', 'hpv'))
```

4.4 assign B/Plasma

[ ]:
```
obj.HNSCC.bp <- obj.HNSCC %>% subset(dCT == 'B/Plasma')
obj.HNSCC.bp

obj.HNSCC.bp <- obj.HNSCC.bp %>%
                NormalizeData(normalization.method = "LogNormalize", scale.
 ↪factor = 10000, verbose = F) %>%
                FindVariableFeatures(selection.method = "vst", nfeatures =␣
 ↪1000, verbose = F) %>%
                ScaleData(vars.to.regress = c("nCount_RNA"), verbose = F) %>%
                RunPCA(verbose = F) %>%
                RunHarmony(group.by.vars = "SampleID", plot_convergence = TRUE)

## cluster
nPC <- min(PC_selection_harmony(obj.HNSCC.bp)$PCselect)
obj.HNSCC.bp <- obj.HNSCC.bp %>%
                 RunUMAP(reduction = "harmony", dims = 1:nPC, umap.method =␣
 ↪"uwot") %>%
                 RunTSNE(reduction = "harmony", dims = 1:nPC) %>%
                 FindNeighbors(reduction = "harmony", dims = 1:nPC) %>%
                 FindClusters(resolution=0.1) %>% FindClusters(resolution=0.2)␣
 ↪%>% FindClusters(resolution=0.3)
colnames(obj.HNSCC.bp@meta.data) <- gsub("RNA_snn_res.0.","r0", colnames(obj.
 ↪HNSCC.bp@meta.data))
colnames(obj.HNSCC.bp@meta.data)
```

[ ]:
```
## marker expression
options(repr.plot.height = 4, repr.plot.width = 5)
marker_list <- c('PTPRC', 'MS4A1', 'CD79A', 'IGKC')
VlnPlot(obj.HNSCC.bp, group.by = 'r03', features = marker_list,
        pt.size = 0, cols = pal_igv("default")(51),
        slot = 'data',assay = 'RNA', raster=FALSE, stack = TRUE, flip = TRUE)+
theme(legend.position = 'none')
```

[ ]:
```
Idents(obj.HNSCC.bp) <- factor(obj.HNSCC.bp$r03, levels = 4:0)
cluster_deg <- FindAllMarkers(obj.HNSCC.bp, assay = "RNA", slot = "data",
                              logfc.threshold = 0.25, min.pct = 0.1, test.use =␣
 ↪"wilcox")
rownames(cluster_deg) <- NULL
```

[ ]:
```
obj.HNSCC.bp@meta.data <- obj.HNSCC.bp@meta.data %>%
                          mutate(dCT = case_when(r03 %in% c('0') ~ 'B',
                                                 r03 %in% c('1', '2') ~␣
 ↪'Plasma',
                                                 r03 %in% c('3', '4') ~␣
 ↪'Others',
```

```
                                                    TRUE ~ 'Others'))
obj.HNSCC.bp@meta.data <- obj.HNSCC.bp@meta.data %>% dplyr::select(-c('r01',␣
↪'r02', 'r03', 'seurat_clusters', 'hpv'))
```

4.5 assign mCT/gCT

[ ]:
```
obj.HNSCC.others <- obj.HNSCC %>% subset(dCT %in% c('T/NK', 'B/Plasma') ==␣
↪FALSE)
obj.HNSCC.others@meta.data <- obj.HNSCC.others@meta.data %>% dplyr::
↪select(-c('hpv'))

obj.HNSCC <- merge(obj.HNSCC.others, c(obj.HNSCC.bp, obj.HNSCC.nkt))
```

[ ]:
```
obj.HNSCC@meta.data <- obj.HNSCC@meta.data %>%
                       mutate(mCT = case_when(dCT %in% c('T_cycling') ~␣
↪'T_proliferation',
                                              dCT %in% c('Plasma', 'B') ~ 'B',
                                              TRUE ~ dCT))
```

[ ]:
```
obj.HNSCC@meta.data <- obj.HNSCC@meta.data %>%
                       mutate(gCT = case_when(mCT %in% c(' T', 'Treg',␣
↪'T_proliferation', 'CD8T', 'B', 'Mast', 'DC', 'Mph') ~ 'Immune',
                                              mCT %in% c('Epi') ~ 'Tumor',
                                              mCT %in% c('Myocyte','Fibro',␣
↪'Endo') ~ 'Stromal',
                                              TRUE ~ 'Others'))
```

4.6 discard unannotated cells

[ ]:
```
obj.HNSCC <- subset(obj.HNSCC, gCT != 'Others')
obj.HNSCC
```

5 UMAP visualization

[ ]:
```
obj.HNSCC <- obj.HNSCC %>%
             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)
## cluster
nPC <- min(PC_selection_harmony(obj.HNSCC)$PCselect)
obj.HNSCC <- obj.HNSCC %>%
```

```
                RunUMAP(reduction = "harmony", dims = 1:nPC, umap.method =␣
 ↪"uwot") %>%
                RunTSNE(reduction = "harmony", dims = 1:nPC)
```

[ ]:
```
options(repr.plot.height = 5, repr.plot.width = 25)
select <- 'umap'
DimPlot_scCustom(obj.HNSCC, pt.size = .1, group.by = "gCT", reduction = select,␣
 ↪label = TRUE, label.size = 4, colors_use = pal_igv("default")(51))|
DimPlot_scCustom(obj.HNSCC, pt.size = .1, group.by = "mCT", reduction = select,␣
 ↪label = TRUE, label.size = 4, colors_use = pal_igv("default")(51))|
DimPlot_scCustom(obj.HNSCC, pt.size = .1, group.by = "dCT", reduction = select,␣
 ↪label = TRUE, label.size = 4, colors_use = pal_igv("default")(51))|
DimPlot_scCustom(obj.HNSCC, pt.size = 1, group.by = "oCT", label = TRUE, label.
 ↪size = 4, colors_use = pal_igv("default")(51))
```

6 save

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