# DE_multiple_condition

*Changhua Yu*

*7/23/2019*

## A Demonstration for Input with 4 different conditions for ShinyApp inspection based on injured/non-injured - KI/WT CD45 positive samples

### Reading in required package and R scripts

```
library(dplyr)
library(Seurat)
library(purrr)
library(cowplot)
library(parallel)
library(roxygen2)
library(reshape2)
library(tibble)
source('../R/cell_cytometry.R')
source('../R/DE_analysis.R')
```

### Parse in 10X Genomics Input

```
path1 = "../../MAP3K3/data/WNI45/"
path2 = "../../MAP3K3/data/KINI45/"
path3 = "../../MAP3K3/data/WTI45/"
path4 = "../../MAP3K3/data/KII45/"

wni = Read10X(data.dir = path1)
kini = Read10X(data.dir = path2)
wti = Read10X(data.dir = path3)
kii = Read10X(data.dir = path4)
```

### Create Seurat Object

```
wni = CreateSeuratObject(counts = wni, project = 'wni', min.cells = 3, min.features = 200)

kini = CreateSeuratObject(counts = kini, project = 'kini', min.cells = 3, min.features = 200)

wti = CreateSeuratObject(counts = wti, project = 'wti', min.cells = 3, min.features = 200)

kii = CreateSeuratObject(counts = kii, project = 'kii', min.cells = 3, min.features = 200)
```

**Specify the "stim" field in the object with a specific condition**

```
wni$stim = "wni"

kini$stim = "kini"

wti$stim = "wti"

kii$stim = "kii"
```

**Calculate the Mitochondria read percentage for pre-processing**

```
wni[["percent.mt"]] <- PercentageFeatureSet(wni, pattern = "^MT-|^mt")

kini[["percent.mt"]] <- PercentageFeatureSet(kini, pattern = "^MT-|^mt")

wti[["percent.mt"]] <- PercentageFeatureSet(wti, pattern = "^MT-|^mt")

kii[["percent.mt"]] <- PercentageFeatureSet(kii, pattern = "^MT-|^mt")
```

**Filter out abnormal cells Based on the mt percent distribution and nFeature distribution**

```
wni <- subset(wni, subset = nFeature_RNA > 500 & nFeature_RNA < 6000 & percent.mt < 20)

kini <- subset(kini, subset = nFeature_RNA > 500 & nFeature_RNA < 6000 & percent.mt < 20)

wti <- subset(wti, subset = nFeature_RNA > 500 & nFeature_RNA < 6000 & percent.mt < 20)

kii <- subset(kii, subset = nFeature_RNA > 500 & nFeature_RNA < 6000 & percent.mt < 20)
```

**Normalize count table and find variable features for dimensionality reduction**

```
wni <- NormalizeData(wni, normalization.method = "LogNormalize", scale.factor = 10000)
wni <- FindVariableFeatures(wni, selection.method = "vst", nfeatures = 2500)

kini <- NormalizeData(kini, normalization.method = "LogNormalize", scale.factor = 10000)
kini <- FindVariableFeatures(kini, selection.method = "vst", nfeatures = 2500)

wti <- NormalizeData(wti, normalization.method = "LogNormalize", scale.factor = 10000)
wti <- FindVariableFeatures(wti, selection.method = "vst", nfeatures = 2500)

kii <- NormalizeData(kii, normalization.method = "LogNormalize", scale.factor = 10000)
kii <- FindVariableFeatures(kii, selection.method = "vst", nfeatures = 2500)
```

**Integrate all four samples together**

```
immune.anchors <- FindIntegrationAnchors(object.list = list(wni, kini,wti,kii))

immune.combined <- IntegrateData(anchorset = immune.anchors)
```

**Proceed to the anchored clustering with UMAP algorithm**

```
DefaultAssay(immune.combined) <- "integrated"
# Run the standard workflow for visualization and clustering
immune.combined <- ScaleData(immune.combined, verbose = FALSE)
immune.combined <- RunPCA(immune.combined, npcs = 30, verbose = FALSE)

# t-SNE and Clustering
immune.combined <- RunUMAP(immune.combined, reduction = "pca",dims = 1:10)
immune.combined <- FindNeighbors(immune.combined, reduction = "pca",dims = 1:10)
immune.combined <- FindClusters(immune.combined, resolution = 0.6)
```

**Save the Seurat Object with integration, normalization, and clustering information for later use**

```
save(immune.combined, file= "../data/WTKICD45POS.RData")
```

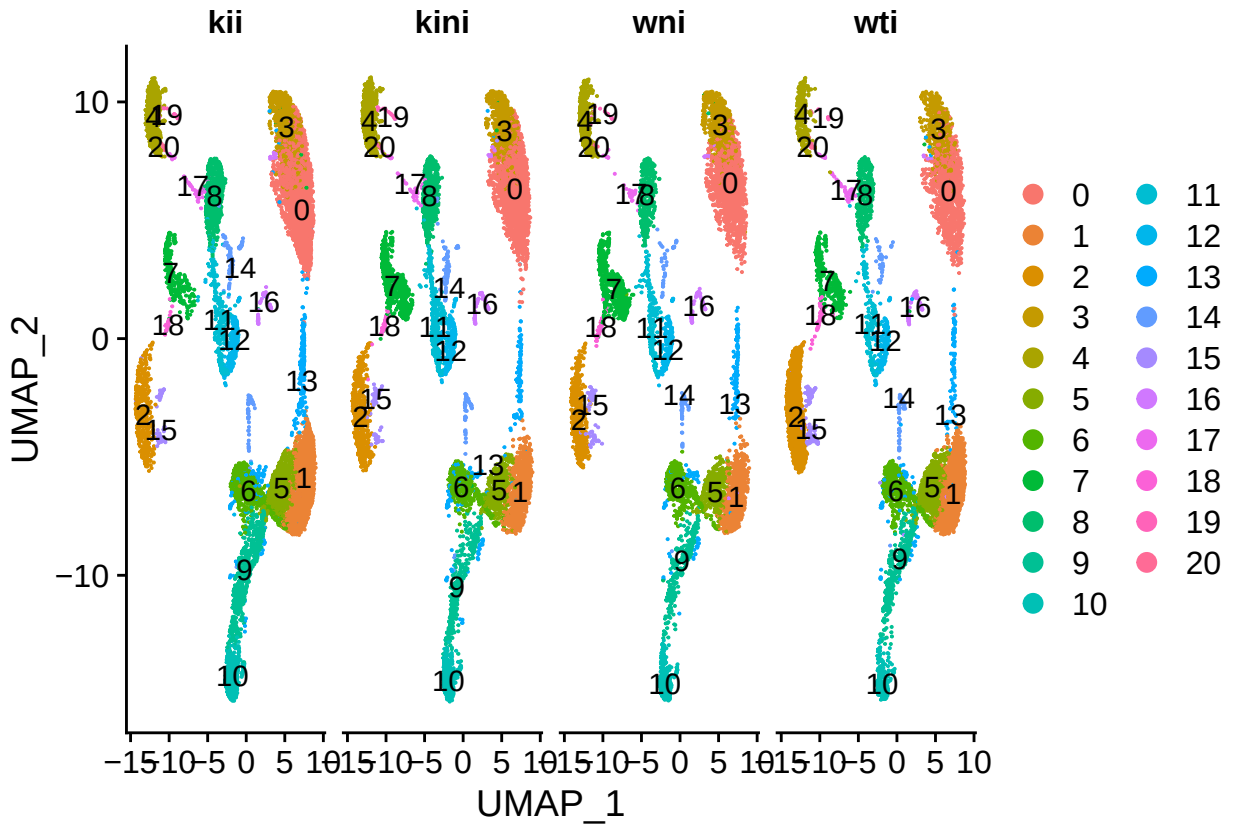
**Load the saved Object back for downstream analysis**

```
load("../data/WTKICD45POS.RData")
```

**Visualizing UMAP clustering**

```
DimPlot(immune.combined, reduction = "umap", split.by = "stim",label = T)
```

```
## Warning: Using `as.character()` on a quosure is deprecated as of rlang 0.3.0.
## Please use `as_label()` or `as_name()` instead.
## This warning is displayed once per session.
```




**Find The Conserve Markers for all clusters**

```
conserved.markers <- Build.ConserveMarkers.All(immune.combined)
```

**Find the differentially expressed gene with a pair of condition specified**

```
diff.genes.wt = DE.Each.Cluster(immune.combined,pair = c("wni","wti"))
diff.gene.ki = DE.Each.Cluster(immune.combined,pair = c("kini","kii"))
diff.genes.i = DE.Each.Cluster(immune.combined,pair = c("wti","kii"))

# Combine the 3 different comparison into a list
diff.genes = list(diff.genes.wt = diff.genes.wt, diff.gene.ki = diff.gene.ki, diff.genes.i = diff.genes
```

**For the clusters within a condition find the representative gene of each cluster**

```
markers.each = Find.Markers.Each(immune.combined,multi = c("kii","kini","wni","wti"))
```

**Output the results for visualization in ShinyApp with multiple conditions**

```
WTKICD45POS.out = Shine.Out(ob = immune.combined, diff = diff.genes, markers.each = markers.each, marke

saveRDS(object = WTKICD45POS.out,file = './ShinyDiff_multi/input/WTKICD45POS_out.rds')
```