R Notebook

Libraries

Whole Dataset

kidney - whole dataset, 10605 samples ## Clean Data

```
# ------- use previously created Seurat object -------
kidney <- readRDS(file = "../../data/kidney_demo_stewart.rds")
```

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

kidney <- subset(kidney, subset = nFeature_RNA > 203 & nFeature_RNA < 7000 & nCount_RNA > 254 & nCount_
```

Transform Data

```
kidney <- SCTransform(kidney, vars.to.regress = "percent.mt", method = "glmGamPoi", verbose = FALSE)
kidney <- RunPCA(kidney, features = VariableFeatures(object = kidney))
```

```
## PC_ 1
## Positive:  GPX3, FTL, ALDOB, SPP1, FXYD2, FTH1, APOE, GSTA1, PDZK1IP1, PCK1
##     FABP1, UGT2B7, MT1G, LDHB, MIOX, SUCLG1, GNLY, PEBP1, NAT8, GATM
##     CYB5A, CXCL14, NKG7, BBOX1, GAPDH, CMBL, GSTA2, CCL5, S100A1, CRYAB
## Negative:  IGFBP5, EMCN, TIMP3, PLAT, IFI27, MGP, CRHBP, RNASE1, PLPP3, SDPR
##     IFITM3, CD74, SLC9A3R2, RAMP2, PLPP1, EPAS1, TGFBR2, FCN3, FLT1, GNG11
##     ENG, SPARC, ESM1, SOST, TM4SF1, PTPRB, HLA-DRB1, ADGRF5, HES1, ID1
## PC_ 2
## Positive:  TMSB4X, GNLY, NKG7, B2M, CCL4, CCL5, TYROBP, GZMB, SRGN, PRF1
##     CXCR4, LYZ, KLRD1, S100A4, FCER1G, ACTB, CD69, CCL3, CTSS, HSPA1B
##     GZMA, KLRB1, JUN, AREG, RGS1, FCGR3A, GZMH, CST7, RPS27, FGFBP2
## Negative:  GPX3, SPP1, ALDOB, FXYD2, FTL, GSTA1, APOE, PCK1, PDZK1IP1, FABP1
##     UGT2B7, CRYAB, SUCLG1, PEBP1, CYB5A, LDHB, GSTA2, DEFB1, CMBL, CXCL14
##     FTH1, NAT8, S100A1, MIOX, MT1G, GATM, CKB, BBOX1, HPD, GAPDH
## PC_ 3
## Positive:  LYZ, AIF1, FCN1, S100A9, LST1, HLA-DRA, CTSS, TYROBP, S100A8, FCER1G
##     CD74, CST3, HLA-DPA1, MNDA, HLA-DPB1, FTL, CSTA, FTH1, MS4A6A, PLAUR
##     HLA-DRB1, HLA-DRB5, NAMPT, VCAN, IL1B, IFI30, SAT1, MS4A7, CD14, CXCL8
## Negative:  GNLY, NKG7, CCL5, CCL4, GZMB, PRF1, KLRD1, FGFBP2, GZMH, GZMA
##     KLRB1, CST7, CTSW, CD69, SPON2, CD7, TRBC1, CD247, KLRF1, CD3D
##     TRAC, IL32, IFNG, IL7R, JUN, B2M, CD2, GZMK, CMC1, HOPX
## PC_ 4
## Positive:  DEFB1, CKB, ATP6V1G3, TMEM213, SPINK1, WFDC2, TACSTD2, CD24, KRT7, ATP6V0D2
##     ATP6V1B1, S100A6, KRT19, CLDN4, MAL, TMEM52B, IGFBP7, ELF3, GDF15, RHCG
##     FXYD2, CA12, KNG1, LGALS3, S100A2, CLCNKB, UMOD, PTGER3, FAM24B, CLCNKA
## Negative:  GPX3, FTL, ALDOB, GNLY, GSTA1, NKG7, FABP1, PDZK1IP1, MT1G, APOE
##     SPP1, UGT2B7, GATM, MIOX, CCL4, NAT8, CCL5, PCK1, GSTA2, B2M
##     GZMB, PRF1, EMCN, BBOX1, HPD, KLRD1, MT1H, PRAP1, FGFBP2, S100A1
```

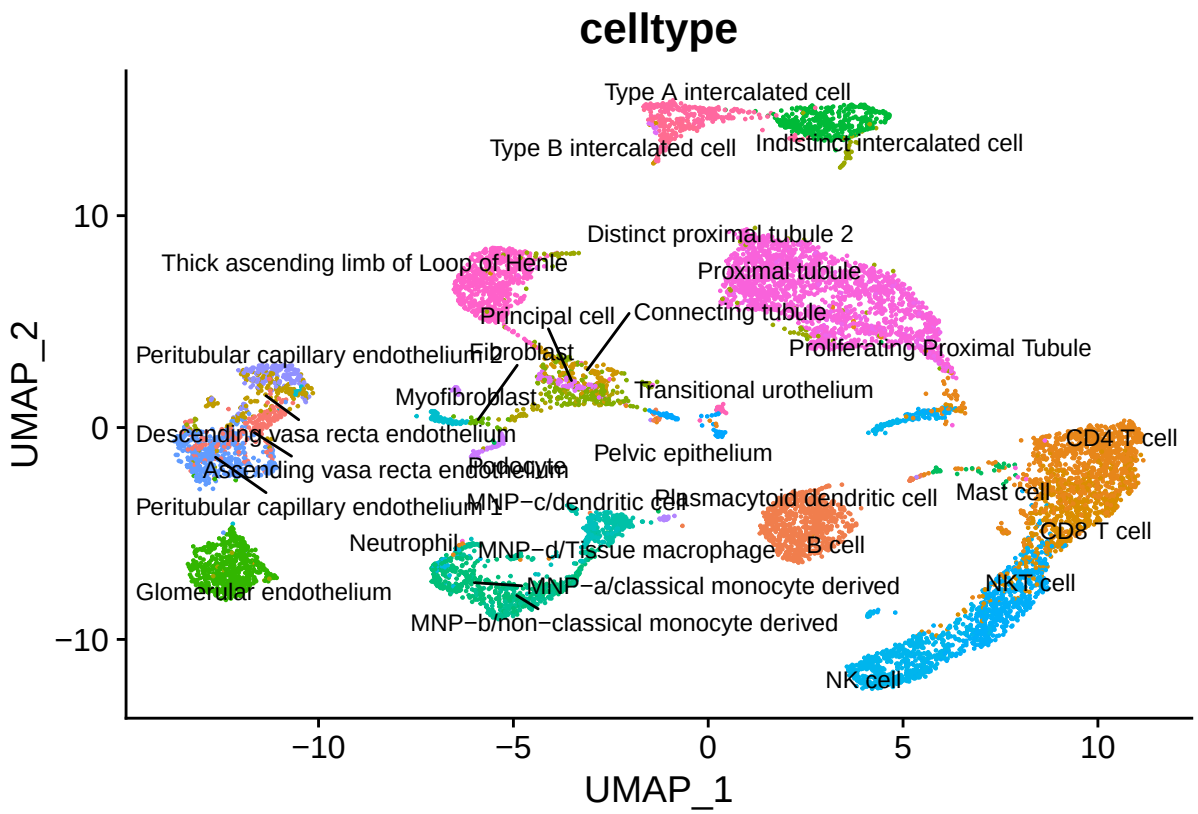
```
## PC_ 5
## Positive:  IGKC, CD74, MS4A1, HSPA1B, CD79A, HLA-DRA, AL928768.3, RPS27, IGLC2, JUN
##     LTB, RP5-887A10.1, IGHA1, IGHG1, BANK1, HLA-DQB1, HLA-DPA1, RPL21, HLA-DQA1, CD79B
##     HLA-DRB1, HLA-DPB1, VPREB3, RPS29, DNAJB1, GPR183, ATP6V1G3, IGLC3, RPS18, HSPA1A
## Negative:  GNLY, NKG7, S100A9, TACSTD2, KRT19, PSCA, S100A6, S100P, SNCG, GZMB
##     ADIRF, PRF1, CLDN4, SPINK1, CCL4, S100A8, FXYD3, SLPI, KLRD1, GDF15
##     ELF3, FGFBP2, SPON2, LCN2, KRT17, UPK2, KRT7, HBB, TYROBP, FCER1G
```

```
kidney <- RunUMAP(kidney, reduction = "pca", dims = 1:10)
```

```
## Warning: The default method for RunUMAP has changed from calling Python UMAP via reticulate to the R-
## To use Python UMAP via reticulate, set umap.method to 'umap-learn' and metric to 'correlation'
## This message will be shown once per session
## 14:37:50 UMAP embedding parameters a = 0.9922 b = 1.112
## 14:37:50 Read 10605 rows and found 10 numeric columns
## 14:37:50 Using Annoy for neighbor search, n_neighbors = 30
## 14:37:50 Building Annoy index with metric = cosine, n_trees = 50
## 0%   10   20   30   40   50   60   70   80   90   100%
## [----|----|----|----|----|----|----|----|----|----|
## **************************************************|
## 14:37:51 Writing NN index file to temp file /tmp/RtmppCRw0S/file4d73c938b91
## 14:37:51 Searching Annoy index using 1 thread, search_k = 3000
## 14:37:55 Annoy recall = 100%
## 14:37:55 Commencing smooth kNN distance calibration using 1 thread
## 14:37:56 Initializing from normalized Laplacian + noise
## 14:37:57 Commencing optimization for 200 epochs, with 427720 positive edges
## 14:38:01 Optimization finished
```

```
DimPlot(kidney, reduction = "umap", group.by="celltype",repel = TRUE, label = TRUE, label.size = 3) + N
```

```
## Warning: ggrepel: 2 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

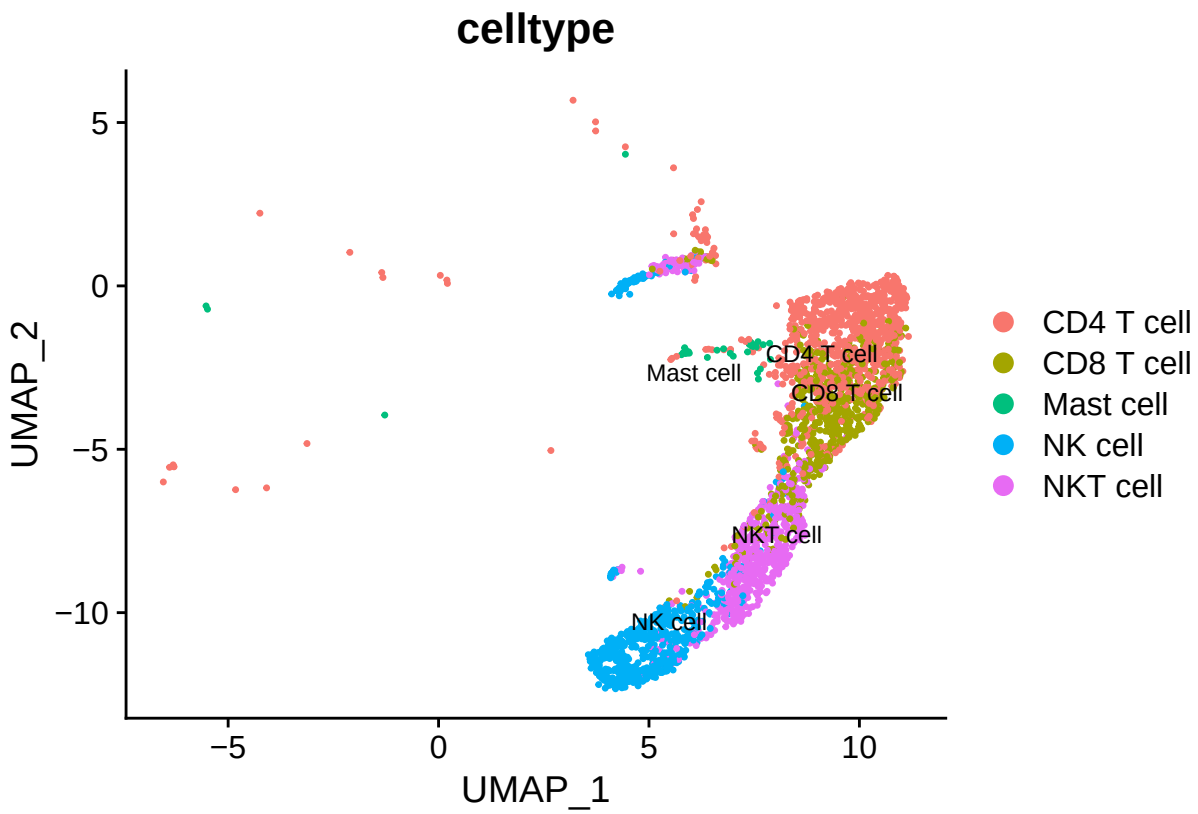


Select Data selected_b – with old (whole dataset) embedding selected_b_2–with new (selected dataset) embedding ## Transform Data

```
to_select <- c("NK cell", "NKT cell", "CD8 T cell", "CD4 T cell", "Mast cell")
selected_meta <- is.element(kidney$celltype, to_select)
kidney <- AddMetaData(kidney, metadata=selected_meta, col.name="selected")
selected_b <- subset(x = kidney, subset = selected == TRUE)
ncol(selected_b)
```

```
## [1] 2901
```

```
DimPlot(selected_b, group.by="celltype",repel = TRUE, label = TRUE, label.size = 3) #+ NoLegend()
```




```
selected_b_2 <- SCTransform(selected_b, vars.to.regress = "percent.mt", method = "glmGamPoi", verbose =
selected_b_2 <- RunPCA(selected_b_2, features = VariableFeatures(object = selected_b_2))
```

```
## PC_ 1
## Positive:  FTL, JUN, LTB, FTH1, HSPA1B, TRAC, MT1G, GPX3, RPL34, IL7R
##     RPS27, GZMK, PDZK1IP1, APOE, FABP1, RPL21, IL32, IGKC, ALDOB, RPS18
##     RPS6, MT1X, RPL13, MT1H, RPS19, HBB, RPL13A, HSPB1, JUNB, NAT8
## Negative:  GNLY, NKG7, CCL4, GZMB, CCL3, PRF1, KLRD1, SPON2, FGFBP2, FCGR3A
##     FCER1G, TYROBP, AREG, KLRF1, CLIC3, CMC1, CCL4L2, PLAC8, CD247, GZMH
##     CD7, CTSW, MYOM2, NEAT1, CST7, KLRB1, S1PR5, SRGN, HAVCR2, HOPX
## PC_ 2
## Positive:  IL7R, CXCR4, RPS29, ZFP36L2, RPS27, RPL41, CD3D, GZMK, TNFAIP3, MT-ND3
##     RGS1, CD8A, BTG1, TSC22D3, XIST, TRAC, CD3G, TOB1, RPS18, CD8B
##     AIM1, RP11-347P5.1, CD3E, MCL1, RPL13, RPL21, CD2, KLF2, RPL13A, CD44
## Negative:  HSPA1A, HBB, FTL, HSPA6, MT1G, GPX3, APOE, PDZK1IP1, GNLY, HSPA1B
##     ALDOB, DNAJB1, FABP1, HSPB1, MT1H, HSP90AA1, FTH1, ADIRF, NAT8, CXCL14
##     HMGCS2, CRYAB, GZMB, PRF1, GATM, BBOX1, JUN, IFNG, FCER1G, CYB5A
## PC_ 3
## Positive:  HBB, APOE, GNLY, GPX3, PDZK1IP1, ALDOB, FTL, GZMB, MIOX, MT1G
##     NKG7, GZMH, S100A8, S100A9, NAT8, GATM, FABP1, MT-CO3, BBOX1, MT1H
##     ADIRF, CXCL14, CD52, RBP5, EEF1A1, CST3, PSCA, PRAP1, SPINK1, MTRNR2L8
## Negative:  JUN, HSPA1B, DNAJB1, HSPA1A, DUSP1, JUNB, FOS, HSP90AA1, NFKBIA, CD69
##     ZFP36, UBC, PPP1R15A, HSP90AB1, IER2, DUSP2, DNAJA1, HSPH1, BTG2, KLF6
##     FOSB, RGS1, MT2A, AREG, EGR1, H3F3B, RGS2, BTG1, MT1X, HSPA8
## PC_ 4
## Positive:  HBB, STMN1, RGS1, HMGB2, HBA2, S100A8, KIAA0101, AREG, TUBB, S100A9
```

```
##     TYMS, TUBA1B, SPINK1, DEFB1, MKI67, HIST1H4C, HBG2, UMOD, XCL2, HLA-DRA
##     TOP2A, NUSAP1, NFKBIA, RRM2, SMC4, CD69, CENPF, HMGN2, PSCA, TK1
## Negative:  FGFBP2, HSPA1B, GNLY, HSPA1A, GZMH, PRF1, HSPA6, NKG7, JUN, PTGDS
##     S100A4, RPS27, GZMB, B2M, SPON2, RPS4Y1, CD52, RPL21, RPL34, CCL5
##     FCGR3A, IL7R, IL32, GIMAP7, LTB, RPS29, RPS6, MYOM2, RPS19, DNAJB1
## PC_ 5
## Positive:  HBB, AREG, HBA2, EEF1A1, TPSAB1, TSC22D3, TPSB2, HPGDS, CD9, NFKBIA
##     TPT1, CD69, CPA3, IL7R, CMC1, MS4A2, KLRB1, ZFP36, NEAT1, GATA2
##     CLU, VWA5A, FTH1, TNFAIP3, S100A8, SPINK1, C1orf186, SRGN, HBA1, GNLY
## Negative:  ACTB, STMN1, TUBA1B, TUBB, KIAA0101, MKI67, HMGB2, TYMS, TOP2A, HIST1H4C
##     NUSAP1, HSPA1B, HMGN2, PCNA, RRM2, UBE2C, GTSE1, CENPF, BIRC5, CKS1B
##     TK1, ASPM, HSPA1A, TMSB4X, PFN1, SMC4, PTTG1, H2AFZ, JUN, MCM7
```

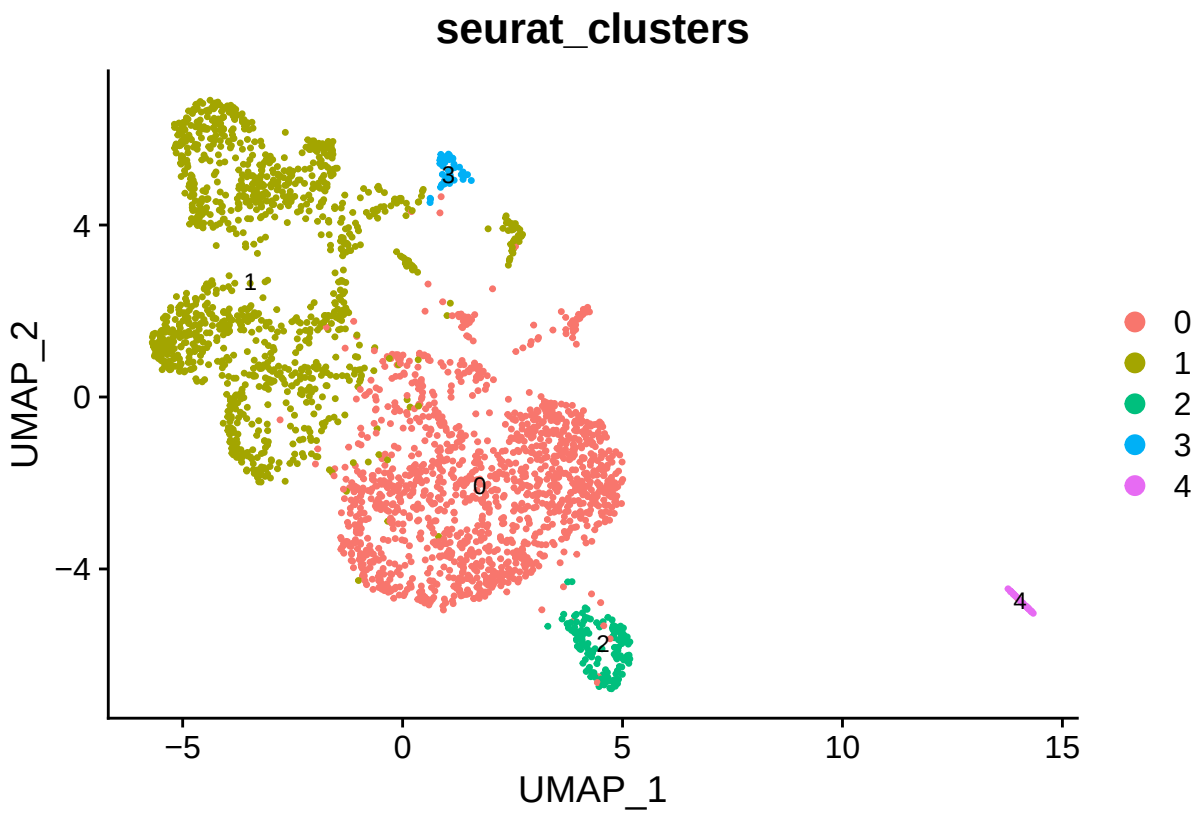
```
n = ncol(selected_b_2)
dim(selected_b_2)
```

```
## [1] 15638  2901
```

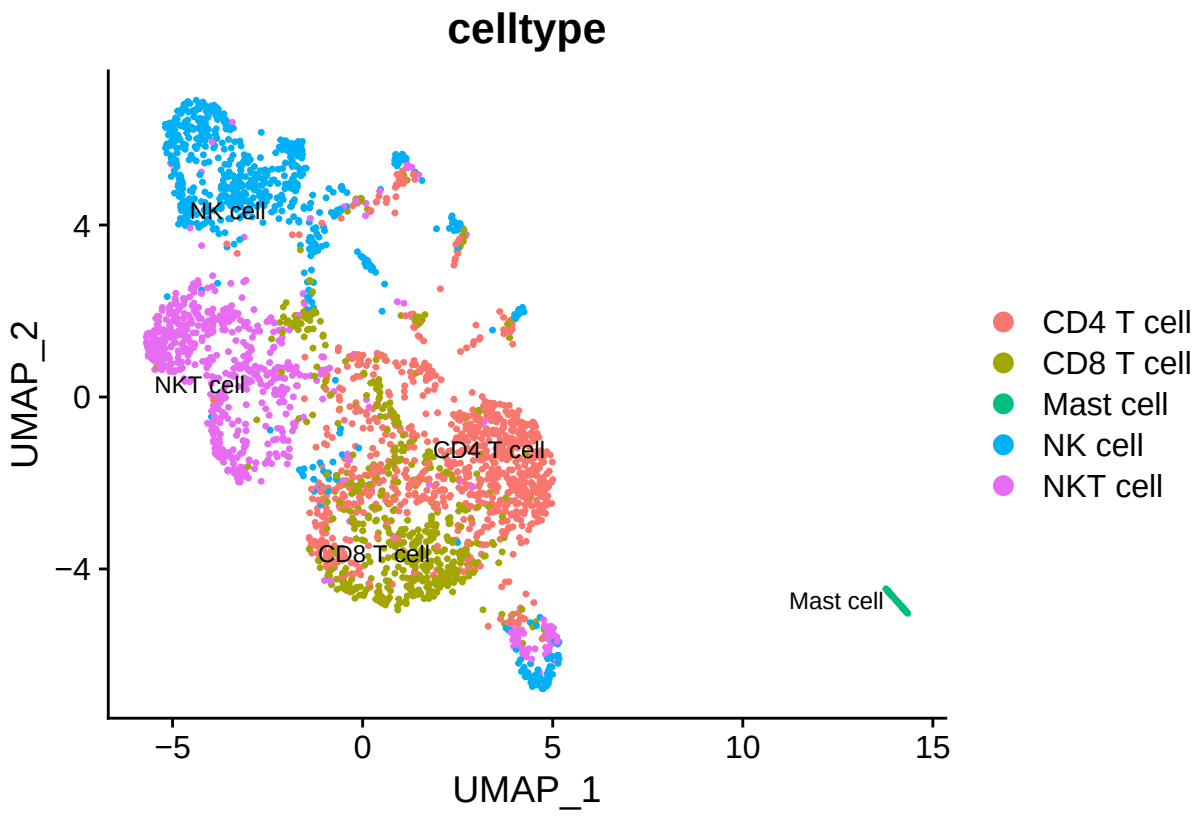
```
type = c("_", "_trimmed_", "_negedges_", "_trimmed_negedges_")
id_type = 2
dim = 30
k = 10
coff = 0.067 #1/15
ord = 10

selected_b_2 <- FindNeighbors(selected_b_2, reduction = "pca", dims = 1:dim, k.param=k, compute.SNN=TRU

selected_b_2 <- FindClusters(selected_b_2, resolution = 0.04, verbose = FALSE)
```

```
selected_b_2 <- RunUMAP(selected_b_2, dims = 1:30, verbose = FALSE)
```

```
DimPlot(selected_b_2, label = TRUE, label.size = 3, group.by="seurat_clusters") #+ NoLegend()
```




```
DimPlot(selected_b_2, group.by="celltype",repel = TRUE, label = TRUE, label.size = 3) #+ NoLegend()
```



Generate SNN graph

```
selected_b_2_snn_temp <- selected_b_2@graphs[["SCT_snn"]]

dim(selected_b_2_snn_temp)
```

```
## [1] 2901 2901
```

```
selected_b_2_snn <- (selected_b_2_snn_temp - diag(nrow=n, ncol=n))
```

```
# ------- Enhance shared edges (may want to repeat multiple times) -------
selected_b_2_pruned_snn_old <- selected_b_2_pruned_snn
for (i in 1:n){
    selected_b_2_pruned_snn[i,] <- selected_b_2_pruned_snn_old[i,]+selected_b_2_pruned_snn_old[,i]
}
```

```
# ------- limitation of nodes degrees -------
for (i in 1:n){
  to_delete <- order(selected_b_2_snn[,i], decreasing = TRUE)[seq(ord+1,n,1)]
  selected_b_2_snn[,i][to_delete] <- integer(n-ord)
  selected_b_2_snn[i,][to_delete] <- integer(n-ord)
}
```

Plot and save graphs

```
library(reticulate)
virtualenv_create("scrna_proj")
```

```
## virtualenv: scrna_proj
```

```
# py_install(c("networkx","matplotlib"), envname = "scrna_proj")
use_virtualenv("scrna_proj")
```

```
import numpy as np
import networkx as nx
from matplotlib import pyplot as plt

id_type, type = int(r.id_type)-1, r.type
n = int(r.n)
k = int(r.k)
ord = int(r.ord)
dim = int(r.dim)

file_name = ''.join(["../../graphs/kidney/", str(n), "_graph_snn", "_k", str(k), "_dim", str(dim), type

G = nx.from_numpy_matrix(r.selected_b_2_snn)
nx.write_gexf(G, file_name)

G = nx.read_gexf(file_name)
pos = nx.spring_layout(G)
plt.cla()
nx.draw_networkx_nodes(G, pos, node_size=10, nodelist=G.nodes)
nx.draw_networkx_edges(G, pos, edgelist=G.edges, style='solid', alpha=0.5, width=1)

file_name = ''.join(["../../graphs/kidney/", str(n), "_graph_snn", "_k", str(k), "_dim", str(dim), type
plt.savefig(file_name, bbox_inches='tight')
```

#Subsampling ## import QA results

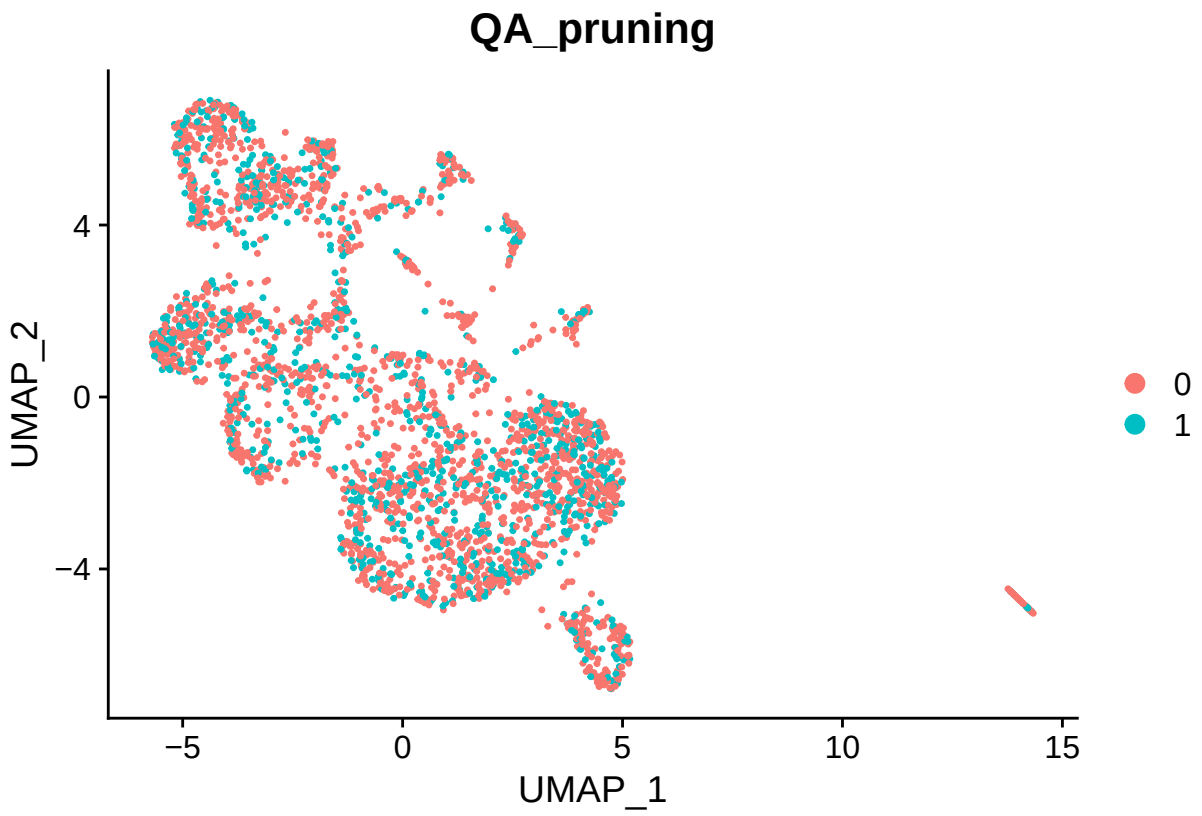
```
# IMPORT GRAPH SUBSAMPLING (indicated by attribute "label1" : 1)
import networkx as nx
QA_output_name = "2901_pru_graph_snn_k10_dim30_trimmed_15v3.gexf"

QA_prune = nx.read_gexf(''.join(["../../dataIn/kidney/", QA_output_name]))
```

```
# GET THE PRUNED NODES
# ------- CHOSE THE CORRECT OPTION BASED ON THE EXPORTED FORMAT -------
pruned = [y[sorted(y.keys())[-1]] for x,y in sorted(QA_prune.nodes(data=True))]
pruned = [y[sorted(y.keys())[-1]] for x,y in QA_prune.nodes(data=True)]
```

Merge MetaData

```
pruned = py$pruned
selected_b_2 <- AddMetaData(selected_b_2, metadata=pruned, col.name="QA_pruning")
```

```
DimPlot(selected_b_2, reduction = "umap", group.by="QA_pruning")
```



Create new graph

```
# PRUNIGN THE GRAPH
pruned = py$pruned
pruned = unlist(pruned)

selected_b_2_pruned <- selected_b_2[,!!pruned]

# ------- Process pruned data set -------
selected_b_2_pruned <- SCTransform(selected_b_2_pruned, method = "glmGamPoi", vars.to.regress = "percen
selected_2_pruned <- RunPCA(selected_b_2_pruned, features = VariableFeatures(object = selected_b_2_prun
```

```
## PC_ 1
## Positive:  FTL, LTB, FTH1, TRAC, JUN, IL7R, RPL34, IL32, RPL21, RPS27
##     MT1G, HSPA1B, RPS6, FABP1, GZMK, GPX3, ALDOB, RPS18, RPS19, PDZK1IP1
##     GPR183, RPS29, RPS4Y1, RPL10, RPL41, JUNB, IGKC, MT1H, APOE, CRYAB
## Negative:  GNLY, NKG7, GZMB, KLRD1, CCL4, PRF1, FCGR3A, SPON2, FGFBP2, TYROBP
##     FCER1G, CCL3, KLRF1, CLIC3, AREG, GZMH, CMC1, PLAC8, CD247, CTSW
##     NEAT1, KLRB1, CD7, CST7, CCL4L2, ZEB2, S1PR5, SRGN, MYOM2, PLEK
## PC_ 2
## Positive:  HSPA1A, FTL, HSPA6, GPX3, HBB, MT1G, PDZK1IP1, FTH1, APOE, ADIRF
##     FABP1, MT1H, ALDOB, HSP90AA1, HSPA1B, CRYAB, DNAJB1, HSPB1, CYB5A, NAT8
##     CXCL14, FCER1G, GNLY, GZMB, GATM, HMGCS2, PRF1, MIOX, BBOX1, ACAA2
## Negative:  CXCR4, IL7R, ZFP36L2, RPS29, TSC22D3, CD8A, MT-ND3, CD3D, TNFAIP3, RPS27
##     RPL41, RPS12, GZMK, CD8B, KLF2, CD3G, AIM1, XIST, TOB1, MCL1
##     BTG1, TXNIP, RP11-347P5.1, TRAC, CD44, RGS1, RPS18, NEAT1, AC092580.4, PPP2R5C
```

```
## PC_ 3
## Positive:  JUN, DNAJB1, DUSP1, HSPA1B, HSPA1A, JUNB, FOS, CD69, HSP90AA1, NFKBIA
##     ZFP36, DUSP2, UBC, IER2, PPP1R15A, HSP90AB1, DNAJA1, HSPH1, RGS1, BTG2
##     H3F3B, KLF6, FOSB, BTG1, RGS2, HSPA8, CXCR4, KLRB1, MT2A, HSPD1
## Negative:  GZMB, GNLY, GZMH, CD52, NKG7, FGFBP2, PDZK1IP1, GPX3, TMSB10, APOE
##     MIOX, KLRG1, S100A4, HBB, RPL21, PRF1, ACTB, FCGR3A, RPS12, RPS6
##     ALDOB, GATM, RBP5, NAT8, MT1G, ITGB1, EEF1A1, IL32, HLA-DPA1, BBOX1
## PC_ 4
## Positive:  RGS1, HBB, HBA2, CXCR4, CCL4, MTRNR2L8, HLA-DRA, HLA-DRB5, APOE, HMGB2
##     MIOX, HLA-DRB1, ZNF302, XCL2, CD69, TNFAIP3, ZFP36L2, XIST, ZFP36, CMC1
##     STMN1, DUSP2, HBA1, CD74, TSC22D3, COTL1, HIST1H4C, PDZK1IP1, KIAA0101, GPX3
## Negative:  HSPA1B, HSPA1A, JUN, DNAJB1, RPS4Y1, LTB, FGFBP2, RPL34, MYOM2, SPON2
##     GIMAP7, RPL21, S100A4, PRF1, RPS6, HSPA6, IL7R, PLAC8, CFAP97, TNF
##     RPS19, RPS27, MBP, NKG7, DOK2, TRAC, ATM, SH3BP5, HSPH1, FAM65B
## PC_ 5
## Positive:  GNLY, GZMH, TSC22D3, KLF2, ANXA1, CXCR4, KLF6, TOB1, HBB, IL7R
##     DUSP1, PATL2, GZMB, HBA2, CD52, TRGC2, TNFAIP3, FGFBP2, EEF1A1, RPS12
##     MCL1, HLA-DPB1, KLRG1, ITGB1, CD3G, CD8A, AC016831.7, PPP2R5C, ZFP36, NKG7
## Negative:  ACTB, STMN1, GZMK, TUBA1B, FCER1G, TYMS, TUBB, PTMA, PCNA, HMGN2
##     SH2D1A, SMC4, XCL1, HIST1H4C, ZWINT, MKI67, HMGB2, SMC2, AREG, CKS1B
##     KIAA0101, ALOX5AP, IL2RB, H2AFZ, NUCB2, NUSAP1, TRDC, XCL2, DUT, MCM7
```

```
n = ncol(selected_b_2_pruned)
dim(selected_b_2_pruned)
```

```
## [1] 12446   942
```

```
type = c("_", "_trimmed_", "_negedges_", "_trimmed_negedges_")
id_type = 2
dim = 30
k = 10
coff = 0 #1/15
ord = 10

selected_b_2_pruned <- FindNeighbors(selected_b_2_pruned, reduction = "pca", dims = 1:dim, k.param=k, c
```

```
## Computing nearest neighbor graph
```

```
## Computing SNN
```

```
selected_b_2_pruned_snn <- selected_b_2_pruned@graphs[["SCT_snn"]]

dim(selected_b_2_pruned_snn)
```

```
## [1] 942 942
```

```
selected_b_2_pruned_snn_temp <- selected_b_2_pruned_snn
selected_b_2_pruned_snn <- (selected_b_2_pruned_snn_temp - diag(nrow=n, ncol=n))
remove(selected_b_2_pruned_snn_temp)
# pbmc3k_QA_pruned_snn <- round(pbmc3k_QA_pruned_snn, digits=2)
```

```
# ------- limitation of nodes degrees -------
for (i in 1:n){
  to_delete <- order(selected_b_2_pruned_snn[,i], decreasing = TRUE)[seq(ord+1,n,1)]
  selected_b_2_pruned_snn[,i][to_delete] <- integer(n-ord)
  selected_b_2_pruned_snn[i,][to_delete] <- integer(n-ord)
}
```

```r
# ------- Enhance shared edges (may want to repeat multiple times) -------
selected_b_2_pruned_snn_old <- selected_b_2_pruned_snn
for (i in 1:n){
    selected_b_2_pruned_snn[i,] <- selected_b_2_pruned_snn_old[i,]+selected_b_2_pruned_snn_old[,i]
}
```

Plot and save graphs

```python
import numpy as np
import networkx as nx
from matplotlib import pyplot as plt

id_type, type = int(r.id_type)-1, r.type
n = int(r.n)
k = int(r.k)
ord = int(r.ord)
dim = int(r.dim)

file_name = ''.join(["../../graphs/kidney/", str(n), "pru_graph_snn", "_k", str(k), "_dim", str(dim), ty

G = nx.from_numpy_matrix(r.selected_b_2_pruned_snn)
nx.write_gexf(G, file_name)
G = nx.read_gexf(file_name)
pos = nx.spring_layout(G)
plt.cla()
nx.draw_networkx_nodes(G, pos, node_size=10, nodelist=G.nodes)
nx.draw_networkx_edges(G, pos, edgelist=G.edges, style='solid', alpha=0.5, width=1)

file_name = ''.join(["../../graphs/kidney/", str(n), "pru_graph_snn", "_k", str(k), "_dim", str(dim), ty
plt.savefig(file_name, bbox_inches='tight')
```

```r
knitr::include_graphics(py$file_name)
```