scRNASeq - Seurat

Brandon Hancock

1/26/2021

This page shows how the Seurat analysis was done for this paper

The dataset can be downloaded from this link

Analysis largely derived from the Seurat vignettes

Load the required R packages:

```
library(dplyr)
library(Seurat)
library(patchwork)
library(progress)
library(ggplot2)
library(stringr)
```

Load scRNA seq data from 10x cellranger file, create Seurat Object. The 10x file is the result of running cellranger aggr on our Uninjured and 7D after Injury samples

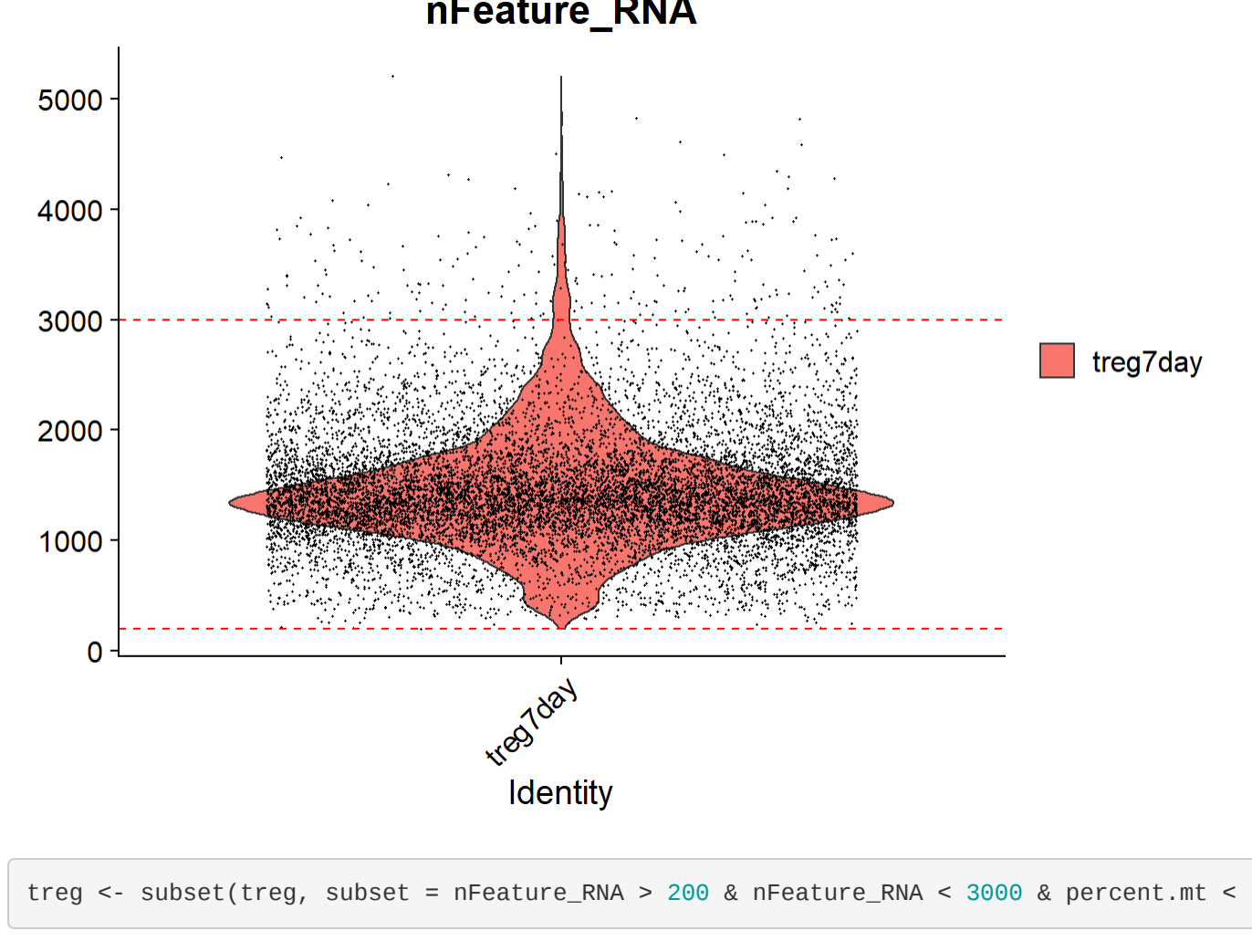
```
treg.data <- Read10X(data.dir = "C:/Users/bh719/Dropbox (Partners HealthCare)/Sally and Jim/FeiData of scRNAseq
 Treg project/10xProcessedData of day 7 after burn/GenomeSeq/outs/filtered_feature_bc_matrix")

treg <- CreateSeuratObject(counts = treg.data,project = 'treg7day',min.cells = 3, min.features = 200)
```

We begin by filtering out low quality cells, see Seurat vignette for more details

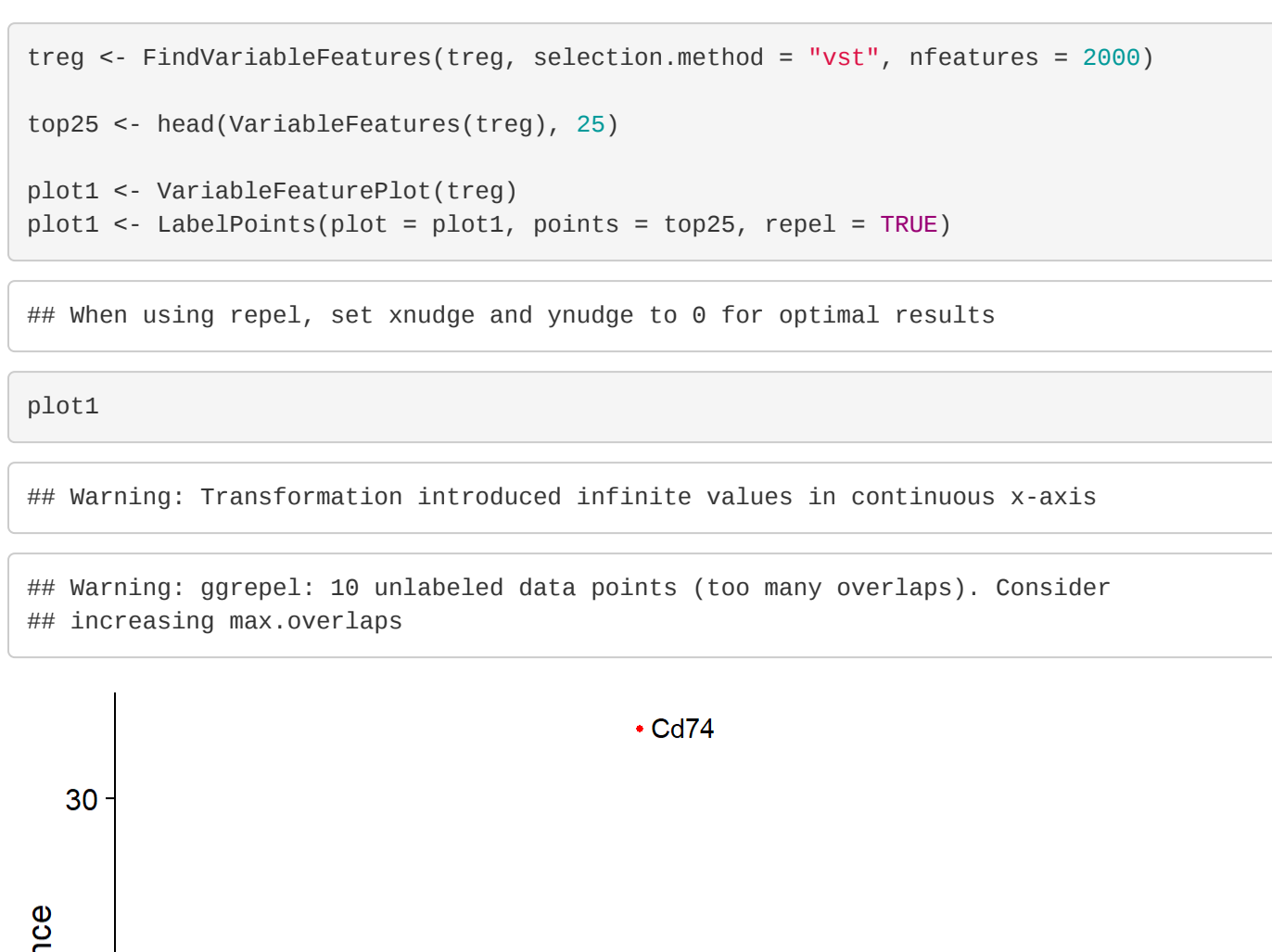
```
treg[["percent.mt"]] <- PercentageFeatureSet(treg, pattern = "^mt-")
```

Visualize cutoffs

```
VlnPlot(treg, features = c("percent.mt")) + geom_hline(yintercept = 6, linetype="dashed", color = "red")
```




```
#lower cutoff actually 200, not 201, but 200 gives error on plot
VlnPlot(treg, features = c("nFeature_RNA")) + geom_hline(yintercept = 201, linetype="dashed", color = "red") + ge
om_hline(yintercept = 3000, linetype="dashed", color = "red")
```




```
treg <- subset(treg, subset = nFeature_RNA > 200 & nFeature_RNA < 3000 & percent.mt < 6)
```

Normalize

```
treg <- NormalizeData(treg)
```

Identify variable features, plot with top 25 labeled

```
treg <- FindVariableFeatures(treg, selection.method = "vst", nfeatures = 2000)

top25 <- head(VariableFeatures(treg), 25)

plot1 <- VariableFeaturePlot(treg)
plot1 <- LabelPoints(plot = plot1, points = top25, repel = TRUE)
```

```
## When using repel, set xnudge and ynudge to 0 for optimal results
```

```
plot1
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning: ggrepel: 10 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Scale the data and run PCA

```
all.genes <- rownames(treg)
treg <- ScaleData(treg, features = all.genes)
```

```
## Centering and scaling data matrix
```

```
#PCA
treg <- RunPCA(treg, features = VariableFeatures(object = treg))
```

```
## PC_ 1
## Positive:  Rpl13, Rps20, Rps19, Rpl35, Rps26, Rpl10a, Rps2, Rpl32, Bcl2, Rps18
##     Rpl15, Rplp0, Eef1a1, Rpl12, Rpl36a, Rpl28, Rpsa, Ccr7, Rpl3, Ly6c1
##     Rplp1, Igfbp4, Sell, Klf2, Satb1, Ms4a4b, Npm1, Rps6, Gm2682, Actn1
## Negative:  S100a6, Icos, S100a11, S100a4, Tigit, Maf, S100a10, Glrx, Itgae, Lgals1
##     Smco4, Capg, Srgn, Tnfrsf1b, Ctla4, Ccr2, Prr13, Ahnak, Klrg1, Tnfrsf4
##     Ikzf2, Rora, Pglyrp1, Rilpl2, AW112010, Psen2, Ttc39c, Tnfrsf18, Itm2c, Ndfip1
## PC_ 2
## Positive:  H2-Ab1, H2-Aa, H2-Eb1, Cd79a, H2-DMb2, Iglc2, Iglc3, Ms4a1, Ly6d, Ebf1
##     Fcmr, Napsa, Mef2c, Ly86, Fcer2a, Tnfrsf13c, Cd19, Cd79b, Bank1, Mzb1
##     Siglecg, Ctsh, Cd74, Igkc, Ighd, Blnk, Cd24a, Blk, Syk, Lyn
## Negative:  Ms4a4b, Rgs10, Izumo1r, Thy1, Rplp0, Trbc2, Cd5, Il2rb, Smc4, Il2ra
##     Ccnd2, Tmsb4x, Ldha, Ifi27l2a, Slfn1, Rpl15, Nsg2, Rgs1, Cd6, Rps18
##     Rplp1, AW112010, Ass1, Socs1, Eef1a1, Rpl32, Tnfrsf18, H2afz, Rpsa, Bcl2
## PC_ 3
## Positive:  Hsp90ab1, Mif, Eif5a, Srm, Tnfrsf9, Ptma, Nhp2, Eif4a1, Pa2g4, Npm1
##     Tnfrsf4, Apex1, Ncl, C1qbp, Rps2, Hspe1, Atp5g1, Ran, Marcksl1, Shmt1
##     Tuba1b, Fbl, Ccr8, Nop16, Set, Srsf2, Nop58, Hspd1, Ranbp1, Phb
## Negative:  Klf2, Malat1, Tsc22d3, Samhd1, Ccr2, Ly6a, Tmsb4x, Ighm, Selenop, Crip1
##     Ifngr1, Fgl2, S1pr1, Rflnb, Ifi209, S100a6, S1pr4, 2810474O19Rik, Glrx, Arl4c
##     Il7r, Cd7, Vim, Ms4a4b, Rasgrp2, Lsp1, Myo1f, S100a4, Ahnak, Socs3
## PC_ 4
## Positive:  Stmn1, Birc5, Pclaf, Ube2c, Cdca3, Ccnb2, Cdca8, Spc24, Ccna2, Mki67
##     Rrm2, Tk1, Cenpm, Cks1b, Asf1b, Tpx2, Rplp0, Hmgb2, Crip1, Top2a
##     Tmsb4x, Cenpe, Cdkn3, Clspn, Hist1h2ag, Gm4316, Cenpf, Rad51, Vim, Tacc3
## Negative:  Zfp36l1, Cd83, Tnfrsf9, Tnfrsf4, Bcl2a1b, Ccr8, Nfkbid, Malat1, Ltb, Nr4a1
##     Nfkbia, Pdcd1, Rgs16, Ephx1, Orai1, Dusp1, Izumo1r, Marcksl1, Relb, Bcl2a1d
##     Rel, Nr4a3, Hivep3, Cd82, Egr2, Egr1, Tbc1d4, Junb, Ier5, Rilpl2
## PC_ 5
## Positive:  Ly6a, Actb, Srm, Ppia, Ly6c1, Ccr2, Gzmb, Eif5a, Crip1, C1qbp
##     Slc7a5, Psme2, Ldha, Pfn1, Hnrnpab, Ranbp1, Hspa9, Timm8a1, Mat2a, S100a6
##     Klrg1, S100a4, Cfl1, Nme2, Fgl2, Apex1, Nolc1, Klf2, Ltb4r1, Gnl3
## Negative:  Rps12, Izumo1r, Eef1a1, Rgs10, Rps26, Mxd4, Rpl32, Rplp1, Capn3, Tbc1d4
##     Nrp1, Gm10076, Rpl28, Rpl3, Rps18, Nsg2, Ephx1, Rpsa, Sh2d1a, Cst7
##     Rpl15, Rpl36a, Smpdl3a, Tiam1, Rpl22l1, Gsta4, 2310001H17Rik, Smc4, Rplp0, H2-Oa
```

Cluster

```
#set nn.method to match default of older version of Seurat, used for the original analysis
treg <- FindNeighbors(treg, dims = 1:20, nn.method = 'rann')
```

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

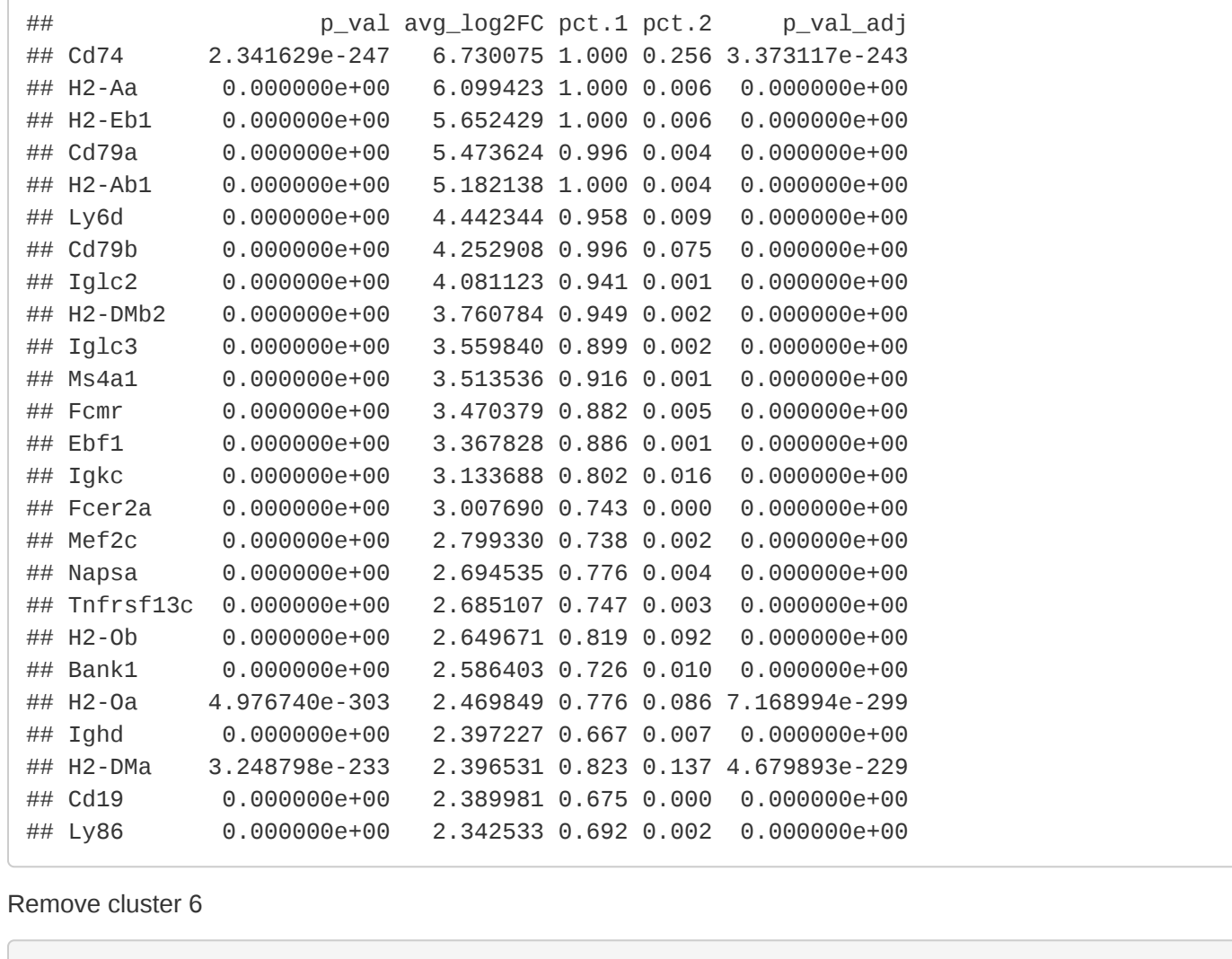
```
## Computing SNN
```

```
treg <- FindClusters(treg, resolution = .5)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 10754
## Number of edges: 335631
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8201
## Number of communities: 11
## Elapsed time: 1 seconds
```

Run TSE dimension reduction

```
treg <- RunTSNE(treg,dims = 1:20)
DimPlot(treg, reduction = "tsne", label = TRUE)
```



We found that cluster 6 expressed B cell markers (Cd74, Cd79, Ms4a1)

```
cluster6.markers <- FindMarkers(treg,ident.1 = 6, min.pct = 0.25)
cluster6.markers <- cluster6.markers[order(cluster6.markers$avg_log2FC,decreasing = TRUE),]
head(cluster6.markers, n = 25)
```

```
##                  p_val avg_log2FC pct.1 pct.2     p_val_adj
## Cd74      2.341629e-247   6.730075 1.000 0.256 3.373117e-243
## H2-Aa      0.000000e+00   6.099423 1.000 0.006  0.000000e+00
## H2-Eb1     0.000000e+00   5.652429 1.000 0.006  0.000000e+00
## Cd79a      0.000000e+00   5.473624 0.996 0.004  0.000000e+00
## H2-Ab1     0.000000e+00   5.182138 1.000 0.004  0.000000e+00
## Ly6d       0.000000e+00   4.442344 0.958 0.009  0.000000e+00
## Cd79b      0.000000e+00   4.252908 0.996 0.075  0.000000e+00
## Iglc2      0.000000e+00   4.081123 0.941 0.001  0.000000e+00
## H2-DMb2    0.000000e+00   3.760784 0.949 0.002  0.000000e+00
## Iglc3      0.000000e+00   3.559840 0.899 0.002  0.000000e+00
## Ms4a1      0.000000e+00   3.513536 0.916 0.001  0.000000e+00
## Fcmr       0.000000e+00   3.470379 0.882 0.005  0.000000e+00
## Ebf1       0.000000e+00   3.367828 0.886 0.001  0.000000e+00
## Igkc       0.000000e+00   3.133688 0.802 0.016  0.000000e+00
## Fcer2a     0.000000e+00   3.007690 0.743 0.000  0.000000e+00
## Mef2c      0.000000e+00   2.799330 0.738 0.002  0.000000e+00
## Napsa      0.000000e+00   2.694535 0.776 0.004  0.000000e+00
## Tnfrsf13c  0.000000e+00   2.685107 0.747 0.003  0.000000e+00
## H2-Ob      0.000000e+00   2.649671 0.819 0.092  0.000000e+00
## Bank1      0.000000e+00   2.586403 0.726 0.010  0.000000e+00
## H2-Oa     4.976740e-303   2.469849 0.776 0.086 7.168994e-299
## Ighd       0.000000e+00   2.397227 0.667 0.007  0.000000e+00
## H2-DMa    3.248798e-233   2.396531 0.823 0.137 4.679893e-229
## Cd19       0.000000e+00   2.389981 0.675 0.000  0.000000e+00
## Ly86       0.000000e+00   2.342533 0.692 0.002  0.000000e+00
```

Remove cluster 6

```
treg <- subset(treg, subset = seurat_clusters != 6)
```

Redo Variable Features

```
treg <- FindVariableFeatures(treg, selection.method = "vst", nfeatures = 2000)
```

Send top 2000 features (genes) to text file for Metascape gene ontology analysis

```
top2000 <- head(VariableFeatures(treg), 2000)
sink("sc2000.txt")
writeLines(unlist(lapply(top2000, paste, collapse=" ")))
sink()
```

Redo Clustering (set resolution high to increase granularity of clusters for later)

```
treg <- RunPCA(treg, features = VariableFeatures(object = treg))
```

```
## PC_ 1
## Positive:  Rpl13, Rps20, Rps19, Rpl35, Rps26, Rpl10a, Rps2, Rpl32, Bcl2, Rps18
##     Rpl15, Rplp0, Eef1a1, Rpl12, Rpl36a, Rpsa, Rpl28, Ccr7, Rpl3, Ly6c1
##     Igfbp4, Rplp1, Ms4a4b, Sell, Klf2, Satb1, Npm1, Rps6, Gm2682, Actn1
## Negative:  S100a6, Icos, S100a11, S100a4, Tigit, Maf, Glrx, S100a10, Itgae, Lgals1
##     Capg, Smco4, Srgn, Tnfrsf1b, Ccr2, Ctla4, Prr13, Ahnak, Klrg1, Tnfrsf4
##     Rora, Pglyrp1, Rilpl2, Ikzf2, Ttc39c, Psen2, AW112010, Myo1f, Gna15, Ptprcap
## PC_ 2
## Positive:  Klf2, Malat1, Tsc22d3, Samhd1, Ccr2, Ly6a, Tmsb4x, Ighm, Selenop, Crip1
##     Ifngr1, Fgl2, S1pr1, Rflnb, Ifi209, S100a6, 2810474O19Rik, S1pr4, Glrx, Il7r
##     Cd7, Arl4c, Kif21b, Rasgrp2, Myo1f, Ahnak, Lsp1, Vim, Socs3, S100a4
## Negative:  Mif, Eif5a, Hsp90ab1, Srm, Tnfrsf9, Ptma, Nhp2, Eif4a1, Pa2g4, Tnfrsf4
##     C1qbp, Npm1, Ncl, Apex1, Rps2, Atp5g1, Ran, Hspe1, Shmt1, Marcksl1
##     Tuba1b, Fbl, Ccr8, Nop16, Set, Srsf2, Nop58, Hspd1, Ranbp1, Nme2
## PC_ 3
## Positive:  Stmn1, Birc5, Pclaf, Ube2c, Cdca3, Ccnb2, Cdca8, Spc24, Ccna2, Mki67
##     Tk1, Rrm2, Cenpm, Cks1b, Asf1b, Tpx2, Hmgb2, Rplp0, Top2a, Crip1
##     Clspn, Cdkn3, Hist1h2ag, Cenpe, Cenpf, Gm4316, Rad51, Cdk1, Tmsb4x, Tacc3
## Negative:  Zfp36l1, Tnfrsf9, Cd83, Tnfrsf4, Bcl2a1b, Ccr8, Nfkbid, Ltb, Malat1, Nfkbia
##     Nr4a1, Rgs16, Pdcd1, Orai1, Ephx1, Izumo1r, Dusp1, Relb, Marcksl1, Bcl2a1d
##     Cd82, Hivep3, Nr4a3, Rel, Rilpl2, Egr2, Egr1, Junb, Tbc1d4, Mir155hg
## PC_ 4
## Positive:  Izumo1r, Rps12, Eef1a1, Nrp1, Rps26, Mxd4, Rpl32, Capn3, Tbc1d4, Rplp1
##     Gm10076, Nsg2, Sh2d1a, Ephx1, Rpl3, Rpl28, Rps18, Rpsa, Cst7, Smpdl3a
##     Tiam1, Rpl15, Smc4, Rpl22l1, Rpl36a, 2310001H17Rik, Gsta4, Ikzf2, Itgb1, Slamf6
## Negative:  Ly6a, Srm, Ccr2, Actb, Ppia, Ly6c1, Gzmb, Eif5a, Crip1, C1qbp
##     Slc7a5, S100a6, Lgals3, S100a4, Klrg1, Fgl2, Psme2, Hspa9, Timm8a1, Ltb4r1
##     Ranbp1, Hnrnpab, Mat2a, Ldha, Nme2, Pfn1, Cxcr6, Nolc1, Apex1, Cfl1
## PC_ 5
## Positive:  Stmn1, Pclaf, Fcer1g, Tyrobp, Spc24, Asf1b, Birc5, Cdca8, Tk1, Rrm2
##     Pdcd1, Ctsh, Ccna2, Tyms, Marcksl1, Tpx2, Cdca3, Cd81, Clspn, Rad51
##     Cd68, Ccnb2, Alox5ap, Csf2rb, Cdk1, Nusap1, Il1r2, Cenps, Cks1b, Cd74
## Negative:  Ifit1, Ifit3, Rtp4, Isg15, Zbp1, Stat1, Ifit3b, Samhd1, Igtp, Ifi203
##     Ifi47, Irf7, Tgtp2, Ifi27l2a, Usp18, Rpl12, Rps18, Rpl32, Rps12, Rps6
##     Ms4a4b, Bst2, Rplp1, Xaf1, Isg20, Iigp1, Mndal, Slfn1, Gbp7, Rsad2
```

```
treg <- FindNeighbors(treg, dims = 1:25)
```

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

```
## Computing SNN
```

```
treg <- FindClusters(treg, resolution = 2.0)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 10517
## Number of edges: 329902
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.6877
## Number of communities: 25
## Elapsed time: 1 seconds
```

```
treg <- RunTSNE(treg,dims = 1:10)
DimPlot(treg, reduction = "tsne", label = TRUE)
```

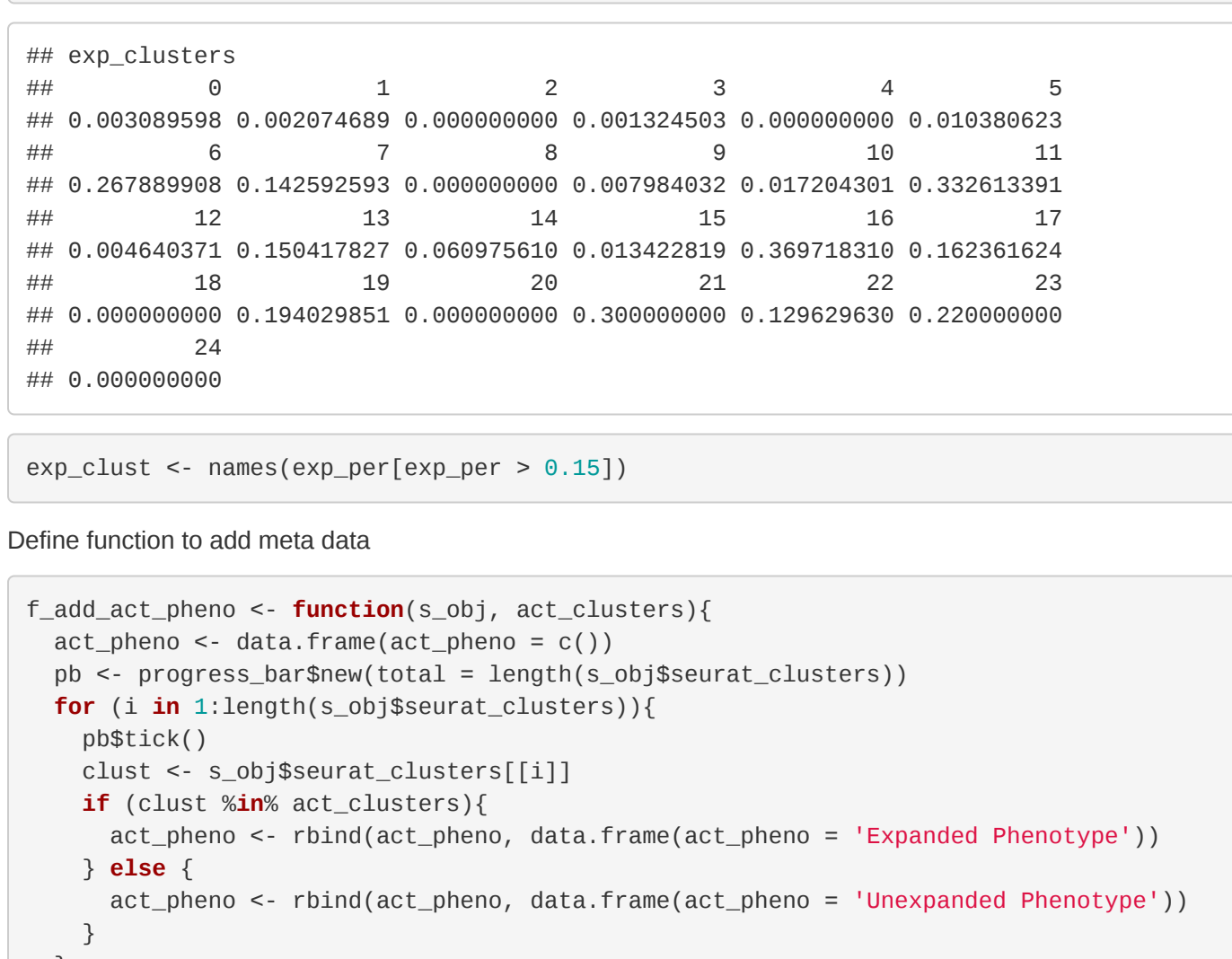
Define functions to add vdj metadata (alpha beta tcr clonotype) and Expanded or NonExpanded classification to cells Function builds on this post

```
add_clonotype <- function(tcr_folder, seurat_obj){
  tcr <- read.csv(paste(tcr_folder,"filtered_contig_annotations.csv", sep=""))
  tcr <- tcr[!duplicated(tcr$barcode), ]

  tcr <- tcr[,c("barcode", "raw_clonotype_id")]
  names(tcr)[names(tcr) == "raw_clonotype_id"] <- "clonotype_id"

  clono <- read.csv(paste(tcr_folder,"clonotypes.csv", sep=""))

  tcr <- merge(tcr, clono[, c("clonotype_id", "cdr3s_aa")])

  tcr <- tcr[, c(2,1,3)]
  rownames(tcr) <- tcr[,1]
  tcr[,1] <- NULL

  clono_seurat <- AddMetaData(object=seurat_obj, metadata=tcr)

  clone_counts <- f_clone_counts(clono_seurat)
  cell_state <- data.frame("Cell State" = c())
  pb <- progress_bar$new(total = length(clono_seurat$clonotype_id))
  for (i in 1:length(clono_seurat$clonotype_id)){
    clonotype_id <- clono_seurat$clonotype_id[[i]]
    cell <- names(clono_seurat$clonotype_id)[[i]]
    pb$tick()
    if (is.na(clonotype_id)){
      cell_state <- rbind(cell_state, data.frame("Cell State" = 'Non Expanded'))
      next
    }
    count <- clone_counts[which(clone_counts$clones == clonotype_id),]$count
    if (count > 1){
      cell_state <- rbind(cell_state, data.frame("Cell State" = 'Expanded'))
    } else {
      cell_state <- rbind(cell_state, data.frame("Cell State" = 'Non Expanded'))
    }
  }
  rownames(cell_state) <- names(clono_seurat$clonotype_id)
  clono_seurat <- AddMetaData(object = clono_seurat, metadata = cell_state)

  return(clono_seurat)
}

f_clone_counts <- function(seurat_object){
  clones <- seurat_object@meta.data$clonotype_id
  clone_counts <- data.frame(clones = c(),count = c())
  for (i in 1:length(clones)){
    c <- clones[[i]]
    if (! clones[[i]] %in% clone_counts$clones){
      clone_counts <- rbind(clone_counts,data.frame(clones = clones[[i]],count = 1))
    } else {
      clone_counts[which(clone_counts$clone == c),]$count <- clone_counts[which(clone_counts$clone == c),]$count
 + 1
    }
  }
  clone_counts <- clone_counts[order(clone_counts$count, decreasing = TRUE),]
  return(clone_counts)
}
```

Add vdj data and plot

```
tcr_folder <- "C:/Users/bh719/Dropbox (Partners HealthCare)/Sally and Jim/FeiData of scRNAseq  Treg project/10xPr
ocessedData of day 7 after burn/12-13-multi/12-13_multi/outs/"
treg <- add_clonotype(tcr_folder,treg)
exp_cells <- WhichCells(treg, expression = Cell.State == 'Expanded')
DimPlot(treg, cells.highlight = exp_cells, reduction = 'tsne',sizes.highlight = 1.5,label.size = 12) + scale_colo
r_manual(labels = c("Non Expanded","Expanded"),values = c("grey","red"))
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```



Designate Expanded cell clusters, use 15% expanded cell representation as min cutoff

```
exp_clusters <- FetchData(treg, vars = c('seurat_clusters'), cells = exp_cells)
x <- table(exp_clusters)
t <- table(treg$seurat_clusters)
exp_per <- x / t
exp_per
```

```
## exp_clusters
##           0           1           2           3           4           5
## 0.003089598 0.002074689 0.000000000 0.001324503 0.000000000 0.010380623
##           6           7           8           9          10          11
## 0.267889908 0.142592593 0.000000000 0.007984032 0.017204301 0.332613391
##          12          13          14          15          16          17
## 0.004640371 0.150417827 0.060975610 0.013422819 0.369718310 0.162361624
##          18          19          20          21          22          23
## 0.000000000 0.194029851 0.000000000 0.300000000 0.129629630 0.220000000
##          24
## 0.000000000
```

```
exp_clust <- names(exp_per[exp_per > 0.15])
```

Define function to add meta data

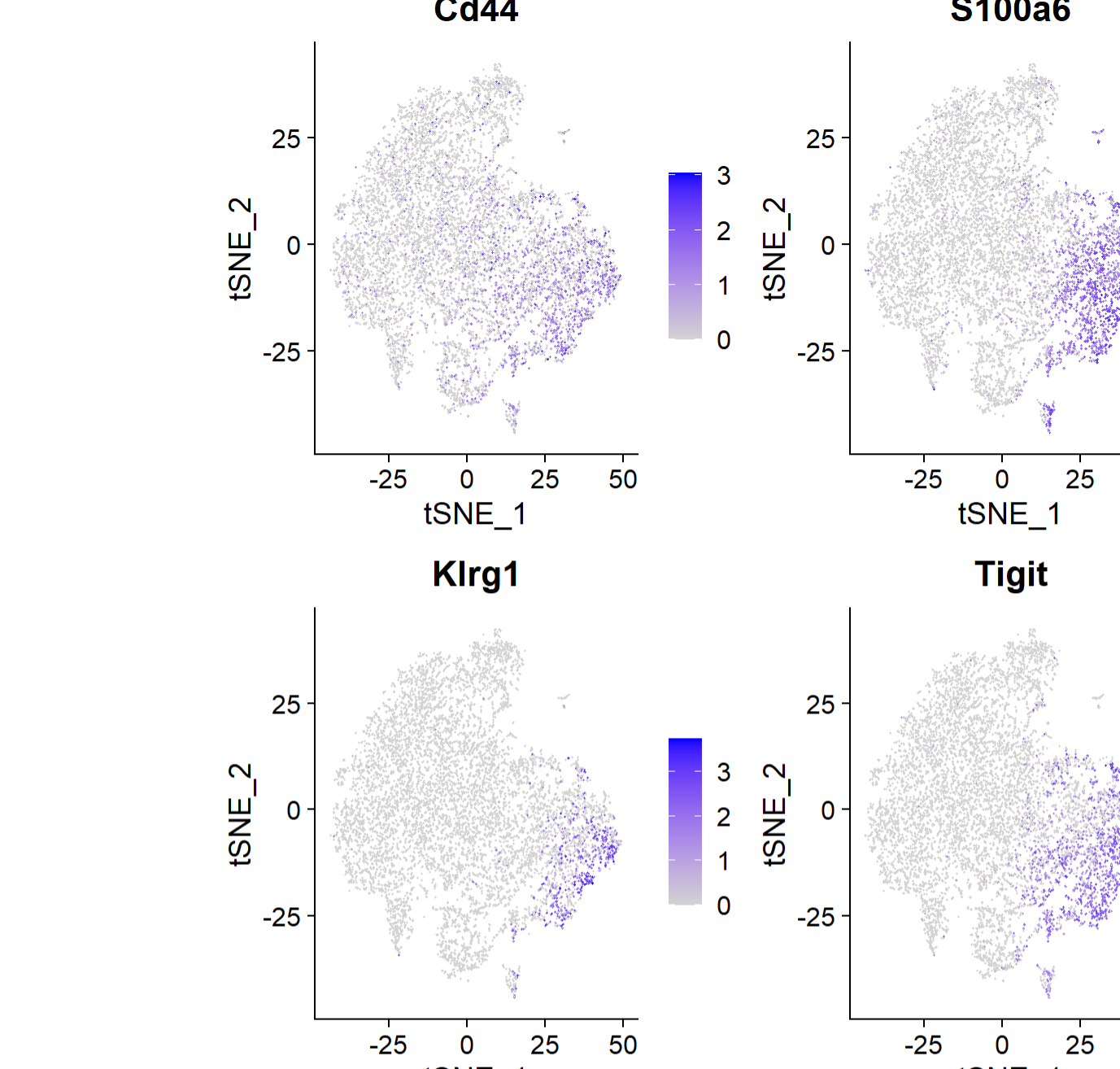
```
f_add_act_pheno <- function(s_obj, act_clusters){
  act_pheno <- data.frame(act_pheno = c())
  pb <- progress_bar$new(total = length(s_obj$seurat_clusters))
  for (i in 1:length(s_obj$seurat_clusters)){
    pb$tick()
    clust <- s_obj$seurat_clusters[[i]]
    if (clust %in% act_clusters){
      act_pheno <- rbind(act_pheno, data.frame(act_pheno = 'Expanded Phenotype'))
    } else {
      act_pheno <- rbind(act_pheno, data.frame(act_pheno = 'Unexpanded Phenotype'))
    }
  }
  rownames(act_pheno) <- names(s_obj$seurat_clusters)
  s_obj <- AddMetaData(object = s_obj, metadata = act_pheno)
}
```

Define function to set plot colors

```
ggplotColours <- function(n = 6, h = c(0, 360) + 15){
  if ((diff(h) %% 360) < 1) h[2] <- h[2] - 360/n
  hcl(h = (seq(h[1], h[2], length = n)), c = 100, l = 65)
}
```

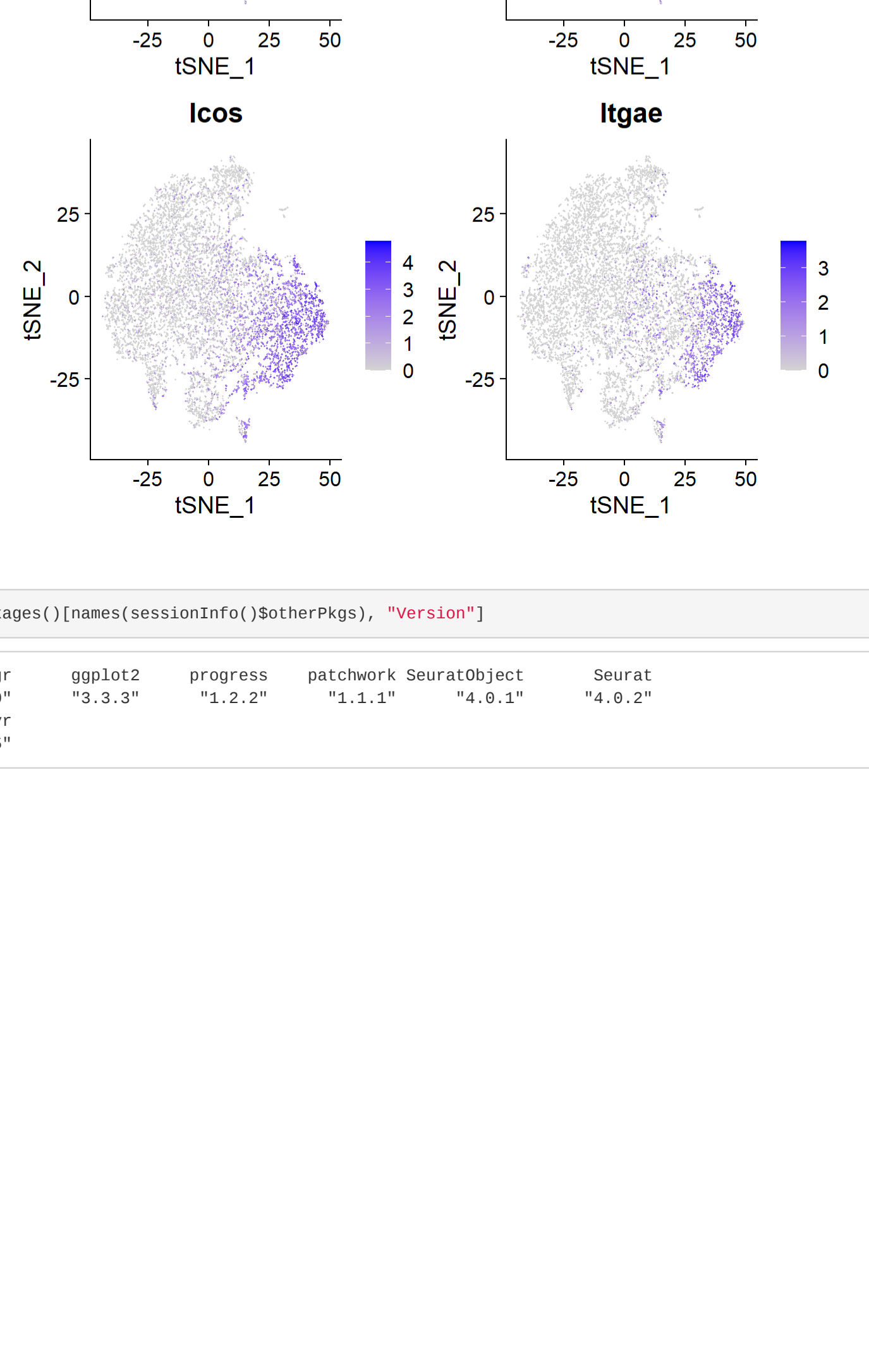
Add meta data and plot

```
#Classify clusters as expanded or non / active or not
treg <- f_add_act_pheno(treg, exp_clust)
Idents(treg) <- treg$act_pheno
DimPlot(treg, reduction = 'tsne',cols = rev(ggplotColours(2)))
```



Feature Plots

```
FeaturePlot(treg, features = c('Cd44','S100a6','Klrg1','Tigit','Icos','Itgae'))
```



Session Info

```
installed.packages()[names(sessionInfo()$otherPkgs), "Version"]
```

```
##      stringr      ggplot2     progress    patchwork SeuratObject       Seurat
##      "1.4.0"      "3.3.3"      "1.2.2"      "1.1.1"      "4.0.1"      "4.0.2"
##        dplyr
##      "1.0.5"
```