Bulk RNA Seq - PCA and Heat Maps

Brandon Hancock

1/15/2021

This page shows how pca plot and heatmaps were generated for this paper

The dataset can be downloaded from this link

Analysis derived from the DESeq2 tutorial

Load the required R packages:

```
library(readxl)
library(readr)
library(DESeq2)
library(pheatmap)
library(ggplot2)
```

Load in the Bulk RNA seq STAR gene counts file:

```
STAR_gene_counts <- read_csv("C:/Users/bh719/Dropbox (Partners HealthCare)/Harvard CyTof/for Brandon/Sally/STAR_Gene_Counts.csv")
```

Remove genes with duplicate entries (1-Mar and 2-Mar)

```
STAR_gene_counts <- STAR_gene_counts[!duplicated(STAR_gene_counts$Gene_ID),]
```

Define Meta Data

```
colmetadat <- data.frame(Injury = c(rep("Uninjured",9),rep("7D after Injury",10)),CD44 = c(rep("CD44 High",4),rep("CD44 Low",5),rep("CD44 High",4),rep("CD44 Low",6)))
row.names(colmetadat) <- colnames(STAR_gene_counts)[2:length(colnames(STAR_gene_counts))]
```

Define DESeq2 matrix

```
gene_row <- STAR_gene_counts$Gene_ID
cmat <- STAR_gene_counts
cmat <- cmat[,!(names(cmat) %in% c('Gene_ID'))]
row.names(cmat) <- gene_row
```

Create DESeq2 object

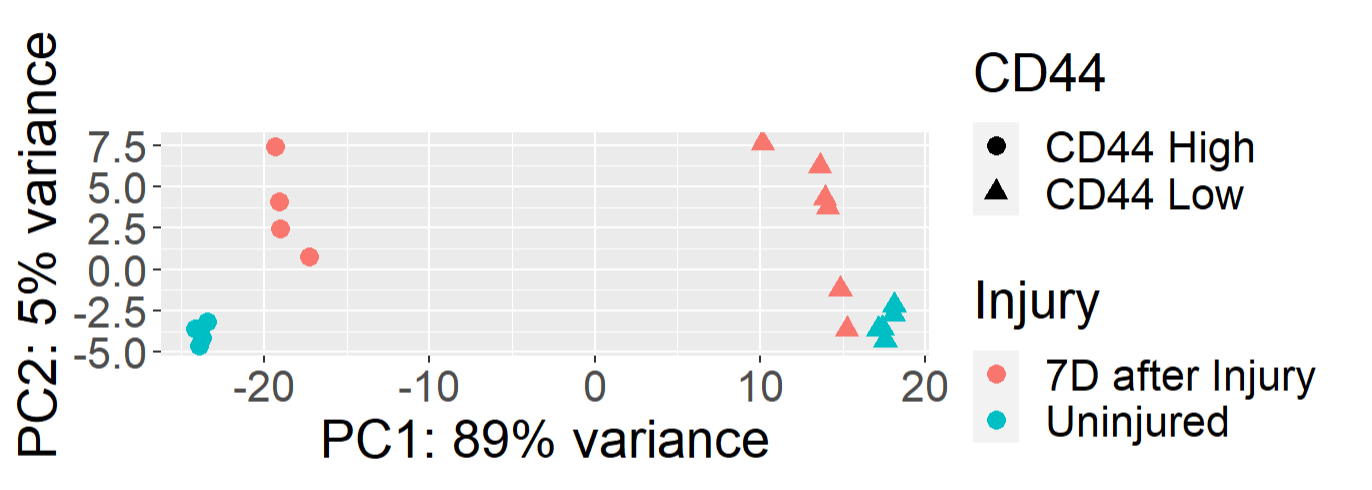
```
dds <- DESeqDataSetFromMatrix(countData = cmat,colData = colmetadat,design = ~ CD44 + Injury)
dds <- dds[rowSums(counts(dds)) >= 10,]
dds$group <- factor(paste0(dds$CD44,dds$Injury))
design(dds) <- ~ group
```

Create the PCA plot

```
vsd <- vst(dds,blind = FALSE)
```

```
##   Note: levels of factors in the design contain characters other than
##   letters, numbers, '_' and '.'. It is recommended (but not required) to use
##   only letters, numbers, and delimiters '_' or '.', as these are safe characters
##   for column names in R. [This is a message, not a warning or an error]
```

```
pcaData <- plotPCA(vsd, intgroup=c("Injury","CD44"), returnData = TRUE)
percentVar <- round(100 * attr(pcaData, "percentVar"))
ggplot(pcaData, aes(PC1, PC2, color = Injury, shape = CD44)) + geom_point(size=3) +
  xlab(paste0("PC1: ",percentVar[1],"% variance")) +
  ylab(paste0("PC2: ",percentVar[2],"% variance")) +
  coord_fixed() +
  theme(text = element_text(size = 20))
```



Define function to calculate variance (stabilized, from vst) of each gene

```
f_get_var <- function(vsd){
  var_list <- c()
  gene_ids <- row.names(vsd)
  for (i in 1:length(gene_ids)){
    gene_row <- as.vector(vsd[gene_ids[i],])
    gene_vec <- c()
    for (j in 1:length(gene_row)){
      gene_vec <- c(gene_vec,gene_row[[j]])
    }
    var_list[gene_ids[i]] <- var(gene_vec)
  }
  return(var_list)
}
```

Get the 2000 genes with the highest variance

```
var_list <- f_get_var(assay(vsd))
var_list <- var_list[order(var_list,decreasing = TRUE)]
gene_list <- head(names(var_list),2000)
```

Use prefixes to find cytokines and cell surface markers from the high variance genes

```
diff_cytokines_cd <- gene_list[grep("^CCL|^CXC|^IFN|^IL|^TNF|CD40LG|FASL|CD70|TGFB|^CD[[:digit:]]",gene_list)]
```

Define a set of genes with known Treg activity

```
treg_genes <- c("TGFB","IL10","ENTPD1","NT5E","LAG3",'TIGIT','CTLA4','ITGAE','KLRG1','ICOS','IL10RA','FGL2','HAVCR2','CD83')
```

Combine lists to create heat map gene list

```
gene_list_heat_cyto <- c(diff_cytokines_cd,treg_genes)
gene_list_heat_cyto <- gene_list_heat_cyto[gene_list_heat_cyto %in% gene_list]
gene_list_heat_cyto <- gene_list_heat_cyto[!duplicated(gene_list_heat_cyto)]
```

Define the heatmap by subsetting genes from the gene count table

```
curr_heat <- STAR_gene_counts[STAR_gene_counts$Gene_ID %in% gene_list_heat_cyto,]
heat_names <- curr_heat$Gene_ID
curr_heat <- curr_heat[,!(names(curr_heat) %in% c('Gene_ID'))]
row.names(curr_heat) <- heat_names
```

```
## Warning: Setting row names on a tibble is deprecated.
```

Define function to scale gene count values and apply to heat map

```
cal_z_score <- function(x){
  (x - mean(x)) / sd(x)
}

curr_heat <- t(apply(curr_heat, 1, cal_z_score))
```

Define function to change capitalization

```
Cap <- function(g){
  g <- paste(toupper(substring(g,1,1)), tolower(substring(g,2)), sep = '')
  return(g)
}
```

Define the gene annotation, 'Treg' or 'Other'

```
treg_annot <- c()
  for (i in 1:length(toupper(gene_list_heat_cyto))){
    if(gene_list_heat_cyto[i] %in% treg_genes){
      treg_annot <- c(treg_annot,'Treg Activity')
    } else {
      treg_annot <- c(treg_annot,"Other")
    }
  }
names(treg_annot) <- gene_list_heat_cyto
names(treg_annot) <- sapply(names(treg_annot), Cap)
gene_row_annot <- data.frame(Treg = treg_annot)
```

Define the sample meta data for the heat map

```
CD44 <- data.frame(CD44 = c(rep("CD44 High",4),rep("CD44 Low",5),rep("CD44 High",4),rep("CD44 Low",6)))
row.names(CD44) <- colnames(curr_heat)
Injury <- data.frame(Injury = c(rep("Uninjured",9),rep("7D after Injury",10)))
row.names(Injury) <- colnames(curr_heat)
sample_col_annot <- cbind(CD44,Injury)
```

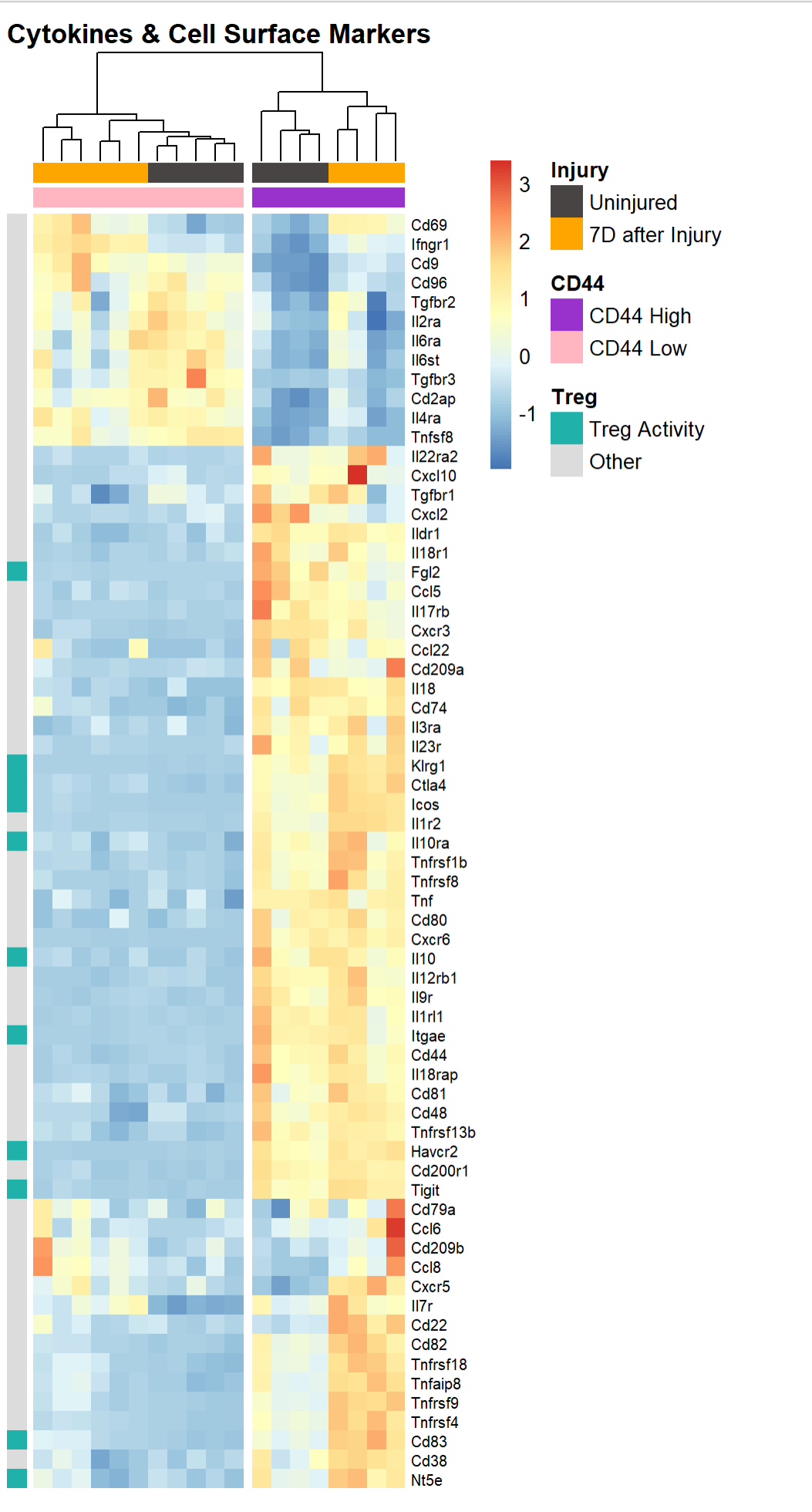
Define colors for meta data

```
annot_colors <- list(Treg = c('Treg Activity' = '#20B2AA','Other' = '#DCDCDC'),
                     Injury = c('Uninjured' = '#474443','7D after Injury' = '#FFA500'),
                     CD44 = c('CD44 High' = '#9932CC', 'CD44 Low' = '#FFB6C1'))
```

Make the heat map

```
row.names(curr_heat) <- sapply(row.names(curr_heat), Cap)
names(gene_row_annot) <- sapply(names(gene_row_annot), Cap)

pheatmap(curr_heat, annotation_row = gene_row_annot,annotation_colors = annot_colors,cutree_cols = 2,annotation_col = sample_col_annot,annotation_names_col = FALSE,annotation_names_row = FALSE,show_colnames = FALSE,fontsize_row = 7.5,main ='Cytokines & Cell Surface Markers',border_color = NA, treeheight_row = 0)
```



For the heat map of transcription factors, we must look up gene ontology information. I use the mygene package here

```
library(mygene)
```

```
## Loading required package: GenomicFeatures
```

```
## Warning: package 'GenomicFeatures' was built under R version 4.0.4
```

```
## Loading required package: AnnotationDbi
```

```
res <- queryMany(gene_list,scopes = 'symbol', fields=c('entrezgene','ensembl.gene','go','description'),species = 'mouse')
```

```
## Querying chunk 1
```

```
## Querying chunk 2
```

```
## Finished
## Pass returnall=TRUE to return lists of duplicate or missing query terms.
```

```
res <- res[!duplicated(res$query),]
```

Define functions to search the mygene query

```
f_getMF <- function(res,gene){
  MF <- res[which(res$query == gene),]$go.MF[[1]]
  return(MF)
}

f_transcription_genes <- function(res,genes){
  tgenes <- c()
  for (i in 1:length(genes)){
    if (length(grep('transcription factor',f_getMF(res,genes[i])$term)) > 0){
      tgenes <- c(tgenes,i)
    }
  }
  return(genes[tgenes])
}
```

Get list of transcription factors with the highest variance (66 genes to match the cytokine heatmap)

```
diff_transcription_facs <- f_transcription_genes(res,gene_list)
diff_transcription_facs <- diff_transcription_facs[1:66]
```

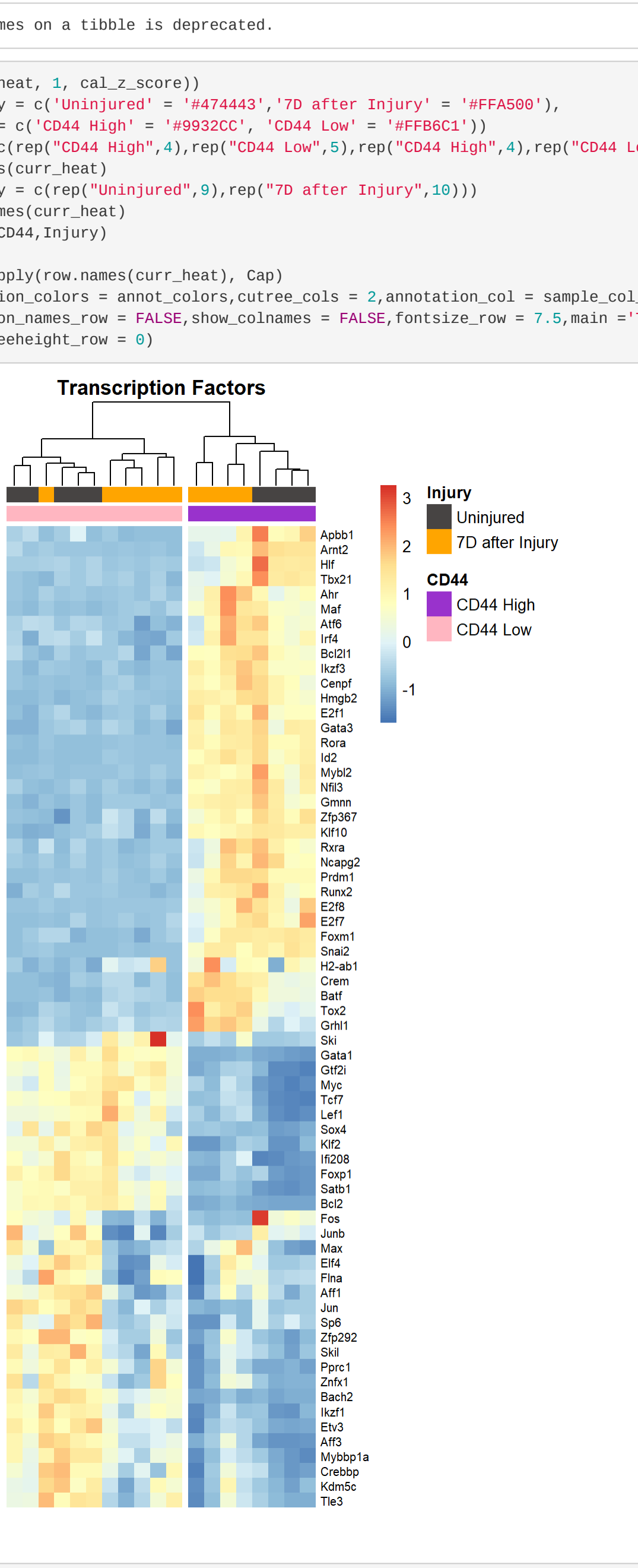
Construct the heatmap as before

```
curr_heat <- STAR_gene_counts[STAR_gene_counts$Gene_ID %in% diff_transcription_facs,]
heat_names <- curr_heat$Gene_ID
curr_heat <- curr_heat[,!(names(curr_heat) %in% c('Gene_ID'))]
row.names(curr_heat) <- heat_names
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
curr_heat <- t(apply(curr_heat, 1, cal_z_score))
annot_colors <- list(Injury = c('Uninjured' = '#474443','7D after Injury' = '#FFA500'),
                     CD44 = c('CD44 High' = '#9932CC', 'CD44 Low' = '#FFB6C1'))
CD44 <- data.frame(CD44 = c(rep("CD44 High",4),rep("CD44 Low",5),rep("CD44 High",4),rep("CD44 Low",6)))
row.names(CD44) <- colnames(curr_heat)
Injury <- data.frame(Injury = c(rep("Uninjured",9),rep("7D after Injury",10)))
row.names(Injury) <- colnames(curr_heat)
sample_col_annot <- cbind(CD44,Injury)

row.names(curr_heat) <- sapply(row.names(curr_heat), Cap)
pheatmap(curr_heat,annotation_colors = annot_colors,cutree_cols = 2,annotation_col = sample_col_annot,annotation_names_col = FALSE,annotation_names_row = FALSE,show_colnames = FALSE,fontsize_row = 7.5,main ='Transcription Factors',border_color = NA, treeheight_row = 0)
```



Session Info

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

```
##              mygene       GenomicFeatures         AnnotationDbi
##            "1.26.0"              "1.42.3"              "1.52.0"
##             ggplot2              pheatmap                DESeq2
##             "3.3.3"              "1.0.12"              "1.30.1"
## SummarizedExperiment              Biobase        MatrixGenerics
##            "1.20.0"              "2.50.0"               "1.2.1"
##          matrixStats         GenomicRanges          GenomeInfoDb
##            "0.58.0"              "1.42.0"              "1.26.7"
##             IRanges             S4Vectors          BiocGenerics
##             "2.24.1"             "0.28.1"              "0.36.1"
##               readr                readxl
##             "1.4.0"               "1.3.1"
```