Test2

Paul Christmann

2022-07-12

The expression of all TRAs associated with a tissue cannot be used to infer organ development

In this research, we attempt to draw conclusions about the developmental state of a tissue based on the expression of genes associated with it alone. Therefore, we analyzed the share of differentially expressed transcripts above a certain expression level over time, as shown in Fig. ???A. Furthermore, we observed trends within the median expression of all differentially expressed transcripts associated with a tissue (Fig. ???B). Since both metrics only showed in miniscule changes, we hypothesized that distinct, counteracting trends in expression exsisted within one tissue. Thus, k-means clustering was used to determine groups of TRAs with similar expression patterns. For each of these clusters, the median expression was plotted as shown in Fig. ???C.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
load("diff_genes_ann_0.01.RData")
load("embryo_df_tissues.RData")
```

```
# Create dataframe with median expression levels for each transcript and week (combining the replicates
embryo_df_tissues_median  = data.frame(Week4 = apply(embryo_df_tissues[1:3], 1, median), Week5 =  apply
embryo_df_tissues_median = cbind(embryo_df_tissues_median, embryo_df_tissues[,c(19,32)])
tissue = "Spleen"
threshold = c(6.6,6.8, 7.0, 7.2, 7.4, 7.6, 8.0)

# Select the data from a tissues
bool_contained = sapply(embryo_df_tissues_median$tissues, function(x) {grepl(tissue, toString(x))})
data_temp = embryo_df_tissues_median[bool_contained,]

#Determine the percentage of expressed genes
count = 0
a = c()
```

```
data_plot = data.frame(Week = c(sapply(c(4:9), function(x) {rep(x, length(threshold))})), Threshold = fa
  for(s in 4:9) {
    for(t in 1:length(threshold)) {
      count = count +1
      a  = c(a,(sum(data_temp[,str_glue("Week", s)] > threshold[t]))/(dim(data_temp)[1]))
    }
  }
data_plot$Percentage= a


#Plot the data
spleen_percentage_plot =  ggplot(data_plot, aes(x = Week, y= Percentage, group = Threshold))   + geom_li
```

```
#Plot of median expression level of all spleen-related genes over time

#Select genes only of target tissue
bool_contained = sapply(diff_genes_ann_0.01$tissues, function(x) {"Spleen" %in% x})

#Graph of mean gene expression over time
spleen_genes = diff_genes_ann_0.01[bool_contained,]
spleen_genes = spleen_genes[, 1:19]
spleen_genes_mean = data.frame(week4=rowMeans(spleen_genes[,1:3]),week5=rowMeans(spleen_genes[,4:6]), we
spleen_genes_graph = apply(spleen_genes_mean, 2, median)
spleen_genes_graph= data.frame (Expression = spleen_genes_graph, Week = c(4:9))

spleen_genes_plot = ggplot(spleen_genes_graph, aes(x = Week, y= Expression))  + geom_line(color = "#000
```

```
#Clustered plot of expression levels over time for the spleen

library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(cluster)
library(stringr)
library(ggplot2)
library(ggsci)
library(grid)
library(gridExtra)
```

```
##
## Attache Paket: 'gridExtra'
```

```
## Das folgende Objekt ist maskiert 'package:dplyr':
##
##     combine
```

```
tissue = "Spleen"

# Functions for later use
expression_difference = function(x,y) {
```

```
  return ((x-y)/((x+y)*0.5))
}

avg_sil = function(k, data) {
  km = kmeans(data, centers = k, nstart = 25)
  ss = silhouette(km$cluster, dist_data)
  mean(ss[, 3])
}

create_plot_data = function(dataset) {
  data_plot = data.frame(Week = c(sapply(c(4:9), function(x) {rep(x, n_cluster)})), Cluster = as.charac
  count = 0
  for(s in 4:9) {
    for(t in 1:n_cluster) {
      count = count + 1
      values = dataset[dataset$Cluster == t, str_glue("Week", s)]
      data_plot[count, "Expression"] = median(values)
      data_plot[count, "Cluster"] = str_glue(data_plot[count, "Cluster"], " [", km$size[t], " genes]")
   }
  }
  return(data_plot)
}



# Select an modify the data used for kmeans
bool_contained = sapply(diff_genes_ann_0.01$tissues, function(x) {grepl(tissue, toString(x))})
data_temp = diff_genes_ann_0.01[bool_contained,]
data_abs = data.frame(Week4 = apply(data_temp[1:3], 1, median), Week5 =  apply(data_temp[4:6], 1, median
                      Week7=apply(data_temp[10:12], 1, median), Week8 =  apply(data_temp[13:15], 1, m
data = data.frame("5_4" = expression_difference(data_abs[,"Week5"], data_abs[,"Week4"]),
                  "6_5" = expression_difference(data_abs[,"Week6"], data_abs[,"Week5"]),
                  "7_6" = expression_difference(data_abs[,"Week7"], data_abs[,"Week6"]),
                  "8_7" = expression_difference(data_abs[,"Week8"], data_abs[,"Week7"]),
                  "9_8" = expression_difference(data_abs[,"Week9"], data_abs[,"Week8"]))
row.names(data) = rownames(data_abs)

# Select optimum cluster number
data = scale(data)
dist_data = get_dist(data, method = "pearson")
n_cluster = which.max(sapply(c(2:10), function(k) {avg_sil(k, data)})) +1

# Create the data for our plot
km = kmeans(data, centers = n_cluster, nstart = 25)
data_plot_raw = cbind(data_abs, Cluster = km$cluster)
plot_data = create_plot_data(data_plot_raw)

# Create the Plot
spleen_clustered_plot =  ggplot(plot_data, aes(x = Week, y= Expression, group = Cluster))  + geom_line(a
          geom_point(aes(colour = Cluster, shape = Cluster)) + labs(title = "Clustered expression") +
          scale_color_npg() + theme_classic()  + ylim(6.5,8.5)
```

```
library(ggplot2)
library(grid)
library(gridExtra)
library(cowplot)
library(ggpubr)
```

```
## Warning: Paket 'ggpubr' wurde unter R Version 4.2.1 erstellt
```

```
##
## Attache Paket: 'ggpubr'
```

```
## Das folgende Objekt ist maskiert 'package:cowplot':
##
##     get_legend
```

```
plot = arrangeGrob(grobs = list(spleen_percentage_plot, spleen_genes_plot, spleen_clustered_plot), nrow=
```

```
## Warning: The shape palette can deal with a maximum of 6 discrete values because
## more than 6 becomes difficult to discriminate; you have 7. Consider
## specifying shapes manually if you must have them.
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```

```
plot_output = as_ggplot(plot) + draw_plot_label(label = c("A", "B", "C"), size = 15, x = c(0.03, 0.35, 
annotate_figure(plot_output, top = text_grob("Expression over time of Spleen-related differentially expr
```

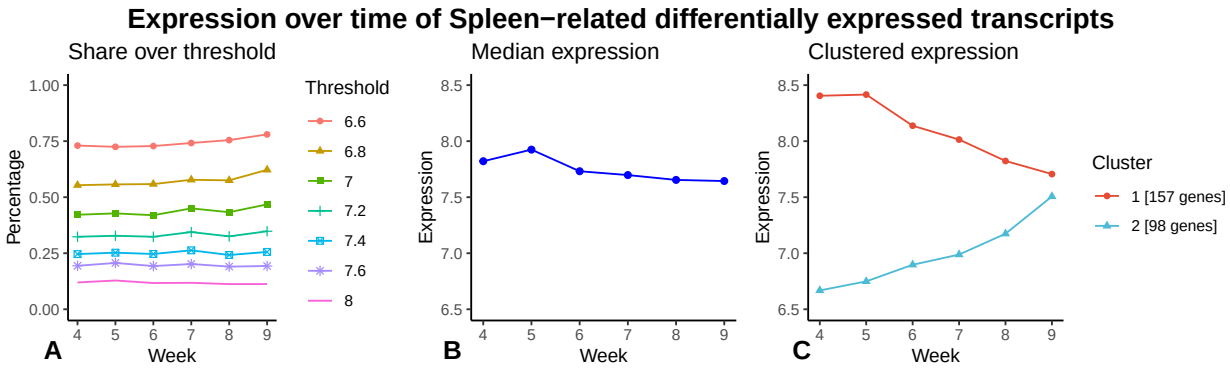

Figure 1: A. For different expression thresholds the share of differentially expressed transcripts with higher expressions than the threshold is depicted. B. The median