Step 2.1: Clustering of gene expression profiles

Carlos Gallardo & Christian Oertlin

17 April, 2023

```
# Import libraries and helper functions
source("code/helper_functions.R")
library(tidyverse)
library(magrittr)
library(patchwork)
library(RColorBrewer)
library(vegan)
library(cluster)
library(ComplexHeatmap)

# Colors
colPals <- vector(mode = "list")
colPals$time <- setNames(c("#FBAA3E", "#2C83BE", "#3EB6BD", "#A3D5B3", "#CD71A8"),
                         nm = c("day0", "day7", "day14", "day21", "day28"))
colPals$time_light <- setNames(c("#FDD6A1", "#A2CDE9", "#AFE2E5", "#DDF0E3", "#E8BDD6"),
                               nm = c("day0", "day7", "day14", "day21", "day28"))
colPals$time_dark <- setNames(c("#D87E04", "#174564", "#1F5C60", "#49A065", "#AA3C7E"),
                              nm = c("day0", "day7", "day14", "day21", "day28"))
colPals$inferno <- c("#000004", "#420A68", "#932667", "#DD513A", "#FCA50A", "#FCFFA4")
colPals$blood_cells <- setNames(c("#E54D34","#77A2D5","#B58B80"),
                                nm = c("granulocytes", "lymphocytes", "monocytes"))
colPals$cell_types <- setNames(c("#83D1F6","#FBAA3E","#FCCA7C","#B58B80","#E54D34",
                                 "#B3177E","#9A509F","#77A2D5","#CAC1DD","#36B449","#C1C1C1"),
                               nm = c("B cell", "Macrophage M1", "Macrophage M2",
                                      "Monocyte", "Neutrophil", "NK cell",
                                      "T cell CD4+ (non-regulatory)", "T cell CD8+",
                                      "T cell regulatory (Tregs)", "Myeloid dendritic cell",
                                      "uncharacterized cell"))
colPals$RdBu <- brewer.pal(11, name = "RdBu")
colPals$biotype <- setNames(c("#395982","#49BED9","#18A38A","#36B449","#826F99",
                              "#9852A5","#FBAA3E","#FCCA7C","#FCFFA4","#C1C1C1"),
                            nm = c("protein_coding", "lncRNA", "miRNA", "snoRNA",
                                   "IG_C_gene", "IG_V_gene", "TR_C_gene",
                                   "TR_J_gene", "TR_V_gene", "other"))
```

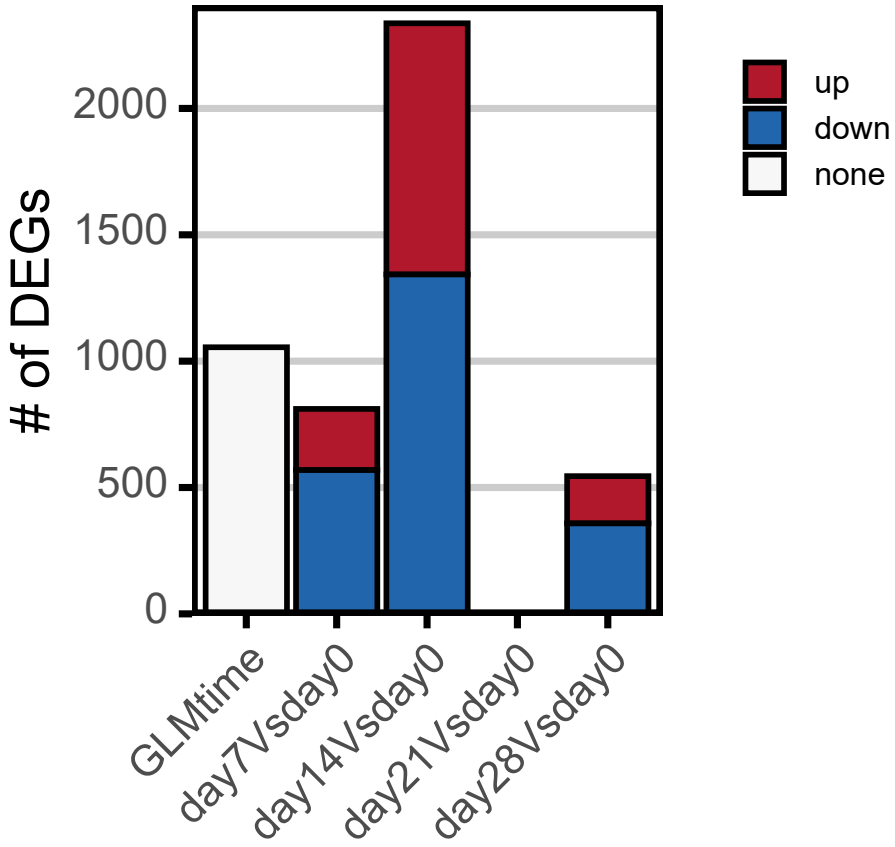
Load data

```
# RNA-seq
RNAseq <- readRDS(file='data/rnaseq/rnaseq_volunteers_9&10_excl.rds')
DESeq2_DEGs <- readRDS(file='data/rnaseq/DESeq2_DEGs_unfilt_volunteers_9&10_excl.rds')
DESeq2_DEGs_filt <- readRDS(file='data/rnaseq/DESeq2_DEGs_filt_volunteers_9&10_excl.rds')
```

Differentially expressed genes (DEGs)

```
# Count up-/downregulated genes for each comparison (di Vs d0)
df <- data.frame(cond = names(DESeq2_DEGs_filt)[1],
                 val = dim(DESeq2_DEGs_filt$GLMtime)[1],
                 type = 'none')

df2 <- lapply(DESeq2_DEGs_filt[-1], function(x) sum(x$log2FoldChange > 0)) %>%
  unlist() %>%
  as.data.frame() %>%
  dplyr::rename(val = '.') %>%
  rownames_to_column(var = 'cond') %>%
  mutate(type = 'up')

df3 <- lapply(DESeq2_DEGs_filt[-1], function(x) sum(x$log2FoldChange < 0)) %>%
```

```
  unlist() %>%
  as.data.frame() %>%
  dplyr::rename(val = '.') %>%
  rownames_to_column(var = 'cond') %>%
  mutate(type = 'down')

df <- df %>%
  bind_rows(df2) %>%
  bind_rows(df3) %>%
  mutate(cond = factor(cond, levels = names(DESeq2_DEGs_filt)),
         type = factor(type, levels = c('up', 'down', 'none')))

ggplot(df, aes(x=cond, y=val, fill=type)) +
  geom_bar(position="stack", stat="identity", color="black", size=1, width=0.9) +
  scale_fill_manual(values = colPals$RdBu[c(2,10,6)]) +
  scale_x_discrete(expand = expansion(mult = c(.15, .15))) +
  scale_y_continuous(
    name="# of DEGs",
    expand =expansion(mult = c(.002, .03))) +
  xlab('') +
  theme_bw(base_size = 20) +
  theme(
    axis.title.y.right = element_text(angle = 90),
    axis.text.x.bottom = element_text(angle = 45, hjust = 1, vjust = 1),
    axis.text.y = element_text(vjust = 0.3),
    legend.title = element_blank(),
    legend.justification=c(0,1),
    panel.grid.major.y = element_line(color = "grey80", linetype = "solid", size = 1.25),
    panel.grid.major.x = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_rect(color = "black", fill = NA, size = 2),
    axis.ticks = element_line(color = "black", size = 1.25),
    legend.position = 'right',
    legend.text = element_text(size=12)
  )
```



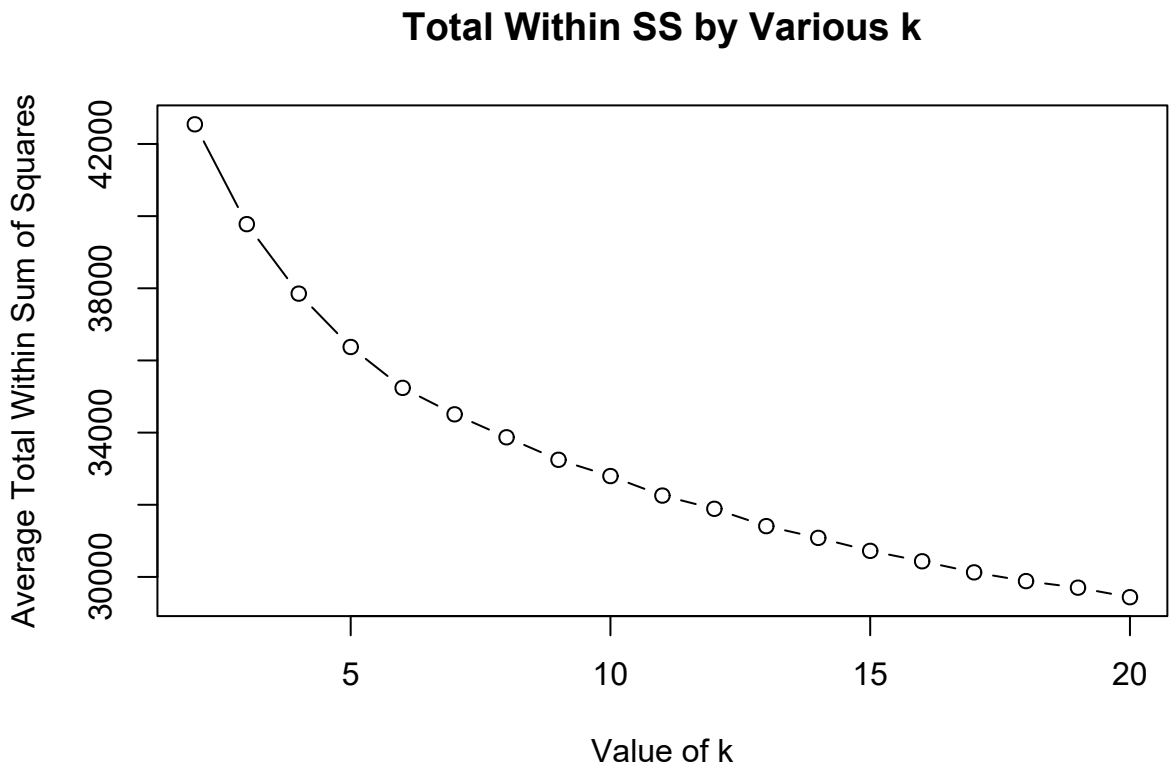
k-means clustering of gene expression over time

```
# Retrieve genes that are differentially expressed in at least one time point
DEGs <- list()
DEGs$Geneid <- lapply(DESeq2_DEGs_filt[2:length(DESeq2_DEGs_filt)], function(x) x$Geneid)
DEGs$GeneSymbol <- lapply(DESeq2_DEGs_filt[2:length(DESeq2_DEGs_filt)], function(x) {
  x$GeneSymbol # keep uniquified gene symbols. To remove: gsub('(.+)_\\d+','\\1',x$GeneSymbol)
})
DEGs <- lapply(DEGs, function(x) x %>% unlist() %>% unique())

# Scale gene expression profiles
rnaseq_scaled <- t(scale(t(RNAseq$filt$DESeq_vst))) %>% as.data.frame()
rnaseq_scaled_DEGs <- rnaseq_scaled[DEGs$GeneSymbol,]
```

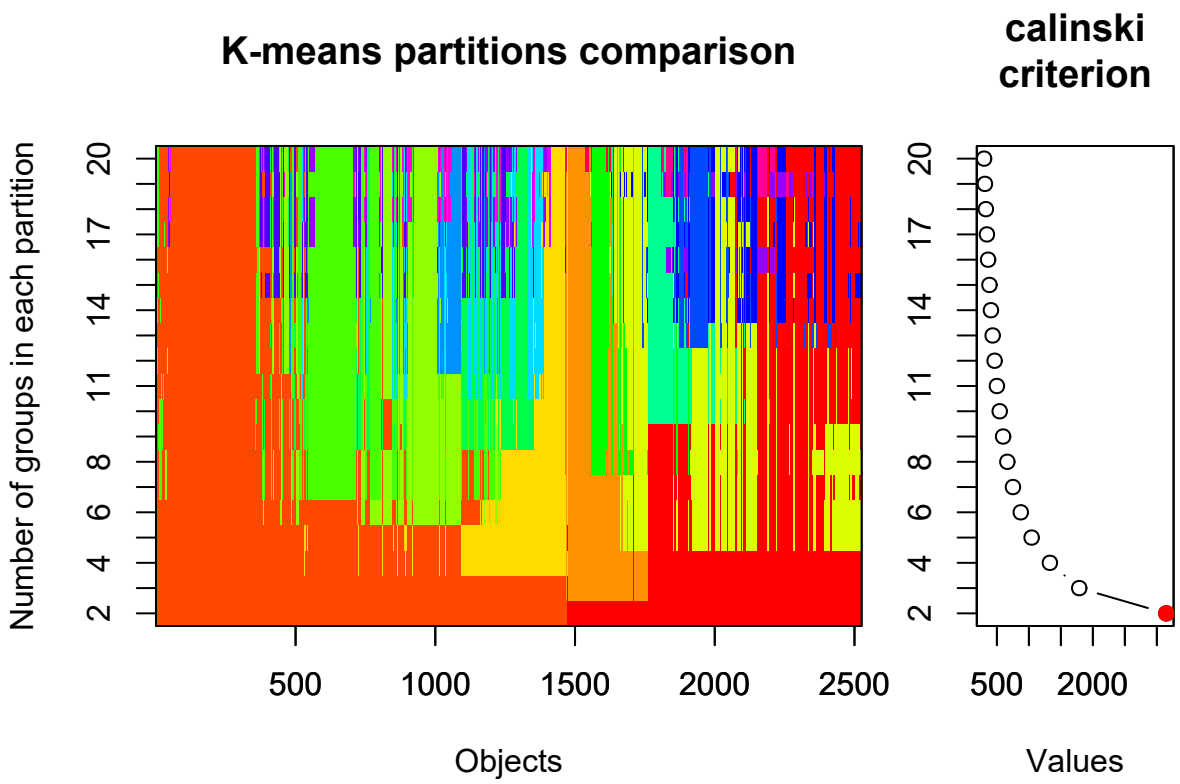
Clustering evaluation

```
# Check within SS at different k
rng<-2:20 #k from 2 to 20
tries <-100 #Run the k Means algorithm 100 times
avg.totw.ss <-integer(length(rng)) #Set up an empty vector to hold all of points
for(v in rng){ # For each value of the range variable
  v.totw.ss <-integer(tries) #Set up an empty vector to hold the 100 tries
  for(i in 1:tries){
    k.temp <-kmeans(rnaseq_scaled_DEGs, centers=v) #Run kmeans
    v.totw.ss[i] <-k.temp$tot.withinss#Store the total withinss
  }
  avg.totw.ss[v-1] <-mean(v.totw.ss) #Average the 100 total withiness
}
```

```
plot(rng,avg.totw.ss,type="b", main="Total Within SS by Various k",
     ylab="Average Total Within Sum of Squares",
     xlab="Value of k")
```

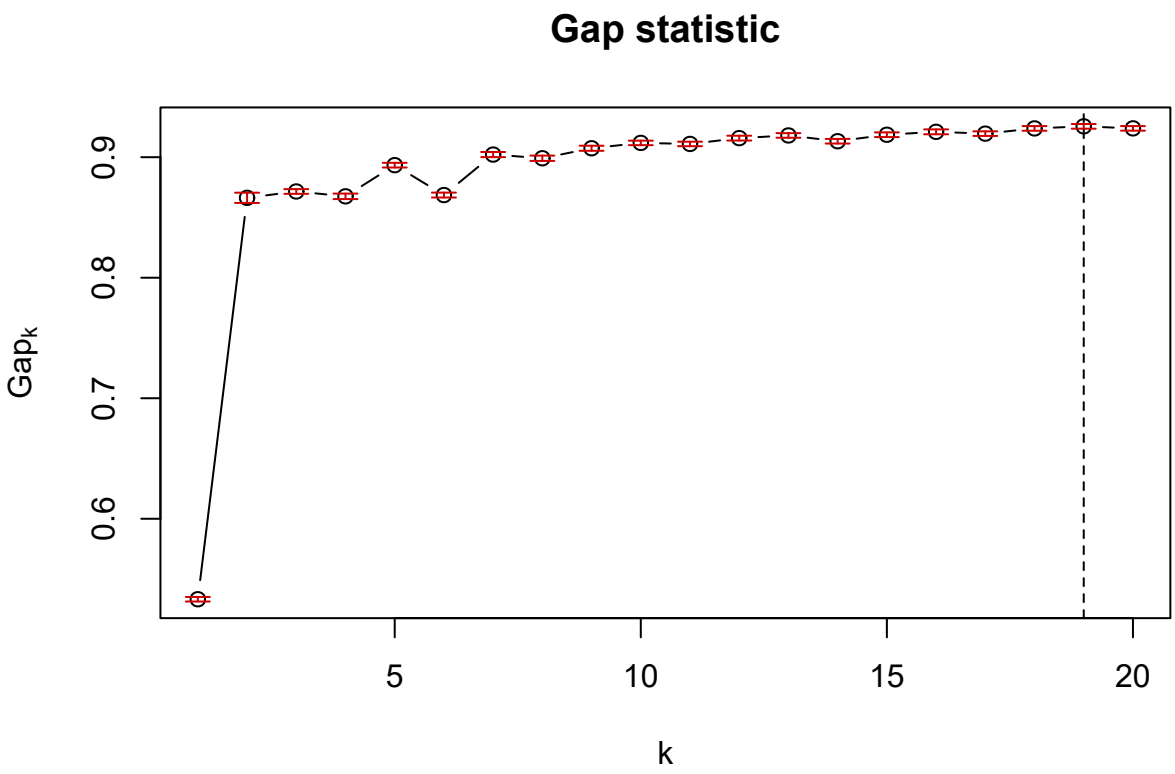



```
# Check k-means partitions
fit <- cascadeKM(rnaseq_scaled_DEGs, 1, 20, iter = 100)
plot(fit, sortg = TRUE, grpmts.plot = TRUE)
```




```
calinski.best <- as.numeric(which.max(fit$results[2,]))
cat("Calinski criterion optimal number of clusters:", calinski.best, "\n")
```

```
## Calinski criterion optimal number of clusters: 2
# Check gap statistic
set.seed(13)
gap <- clusGap(rnaseq_scaled_DEGs, kmeans, 20, B = 100, verbose = interactive())
plot(gap, main = "Gap statistic")
abline(v=which.max(gap$Tab[,3]), lty = 2)
```




```
# Check similarity of cluster profiles with k=4
set.seed(100)
DEGs_clust_k4 <- kmeans(rnaseq_scaled_DEGs, centers = 4)

df <- DEGs_clust_k4$centers %>%
  as.data.frame() %>%
  group.transform(group = RNAseq$filt$design$time,
                  FUN = function(x) apply(x, 1, mean))

cor(t(df))
```

```
##           1         2         3         4
## 1  1.0000000  0.9825767  0.9899959 -0.9953062
## 2  0.9825767  1.0000000  0.9928725 -0.9949472
## 3  0.9899959  0.9928725  1.0000000 -0.9933076
## 4 -0.9953062 -0.9949472 -0.9933076  1.0000000
```

```
# Check similarity of cluster profiles with k=3
set.seed(100)
DEGs_clust_k3 <- kmeans(rnaseq_scaled_DEGs, centers = 3)

df <- DEGs_clust_k3$centers %>%
  as.data.frame() %>%
  group.transform(group = RNAseq$filt$design$time,
                  FUN = function(x) apply(x, 1, mean))

cor(t(df))
```

```
##           1         2         3
## 1  1.0000000 -0.9959542  0.9913322
## 2 -0.9959542  1.0000000 -0.9947338
## 3  0.9913322 -0.9947338  1.0000000
```

```
# Check similarity of cluster profiles with k=2
set.seed(100)
DEGs_clust_k2 <- kmeans(rnaseq_scaled_DEGs, centers = 2)

df <- DEGs_clust_k2$centers %>%
  as.data.frame() %>%
  group.transform(group = RNAseq$filt$design$time,
                  FUN = function(x) apply(x, 1, mean))
```

```
cor(t(df))
```

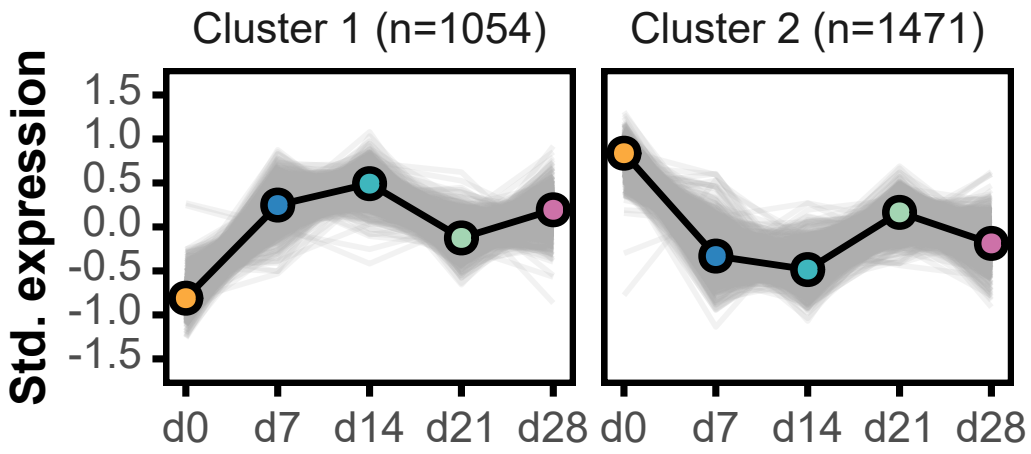
```
##           1          2
## 1  1.0000000 -0.9966174
## 2 -0.9966174  1.0000000
```

Visualize clusters (k=2)

Expression profiles

```
# Average expression profiles per gene (cluster 1 = up, cluster 2 = down)
DEGs_clust_expr <- rnaseq_scaled_DEGs %>%
  group.transform(group = RNAseq$filt$design$time,
                  FUN = function(x) apply(x, 1, mean)) %>%
  rownames_to_column(var = 'GeneSymbol') %>%
  add_column(cluster = recode(.$GeneSymbol, !!!DEGs_clust_k2$cluster), .after = 'GeneSymbol') %>%
  mutate(cluster = recode(cluster, '1'='2', '2'='1'))

# Average expression profiles per cluster (cluster 1 = up, cluster 2 = down)
DEGs_clust_expr_mean <- DEGs_clust_k2$centers %>%
  as.data.frame() %>%
  group.transform(group = RNAseq$filt$design$time,
                  FUN = function(x) apply(x, 1, mean)) %>%
  rownames_to_column(var = 'cluster') %>%
  mutate(cluster = recode(cluster, '1'='2', '2'='1')) %>%
  arrange(cluster)
```

```
df <- DEGs_clust_expr %>%
  pivot_longer(day0:day28, names_to = 'time', values_to = 'expr') %>%
  mutate(cluster = factor(cluster, levels = c('1','2')),
         time = factor(time, levels = names(colPals$time)))

df2 <- DEGs_clust_expr_mean %>%
  pivot_longer(day0:day28, names_to = 'time', values_to = 'expr') %>%
  mutate(cluster = factor(cluster, levels = c('1','2')),
         time = factor(time, levels = names(colPals$time)))

lbls <- c("1" = paste0("Cluster 1 (n=",sum(DEGs_clust_expr$cluster=='1'),")"),
          "2" = paste0("Cluster 2 (n=",sum(DEGs_clust_expr$cluster=='2'),")"))

ggplot(df, aes(time, expr, color=cluster, group=GeneSymbol)) +
  geom_line(size = 1, color=alpha("#AEAEAE", 0.15)) +
  geom_line(data = df2, aes(time, expr, group=cluster), color='black', size = 1.2) +
  geom_point(data = df2, aes(time, expr, group=cluster, fill=time), shape=21, size=3.5, stroke=2, color='black') +
  scale_x_discrete(expand = expansion(mult = c(.06, .06)), labels = c('d0','d7','d14','d21','d28')) +
  scale_y_continuous(limits = c(-1.5,1.5),
                     breaks = seq(-1.5,1.5,0.5),
                     expand = expansion(mult = c(.1, .1))) +
  scale_fill_manual(values = colPals$time) +
  facet_wrap(~cluster, nrow = 1, labeller = as_labeller(lbls)) +
  ylab("Std. expression") +
  xlab("") +
  theme_bw(base_size=20) +
  theme(text = element_text(face = "plain"),
        axis.title.y = element_text(size=18, face = "bold"),
        axis.text.x = element_text(angle = 0, hjust = 0.5, vjust = 0.5),
        axis.text.y = element_text(hjust = 1, vjust = 0.3),
        legend.position = "none",
        axis.ticks = element_line(color = "black", size = 1.25),
        axis.ticks.length = unit(1.5, 'mm'),
        panel.border = element_rect(color = "black", fill = NA, size = 2),
        panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank(),
        panel.grid.major.y = element_blank(),
        panel.grid.minor.y = element_blank(),
        strip.background = element_blank()
        )
```



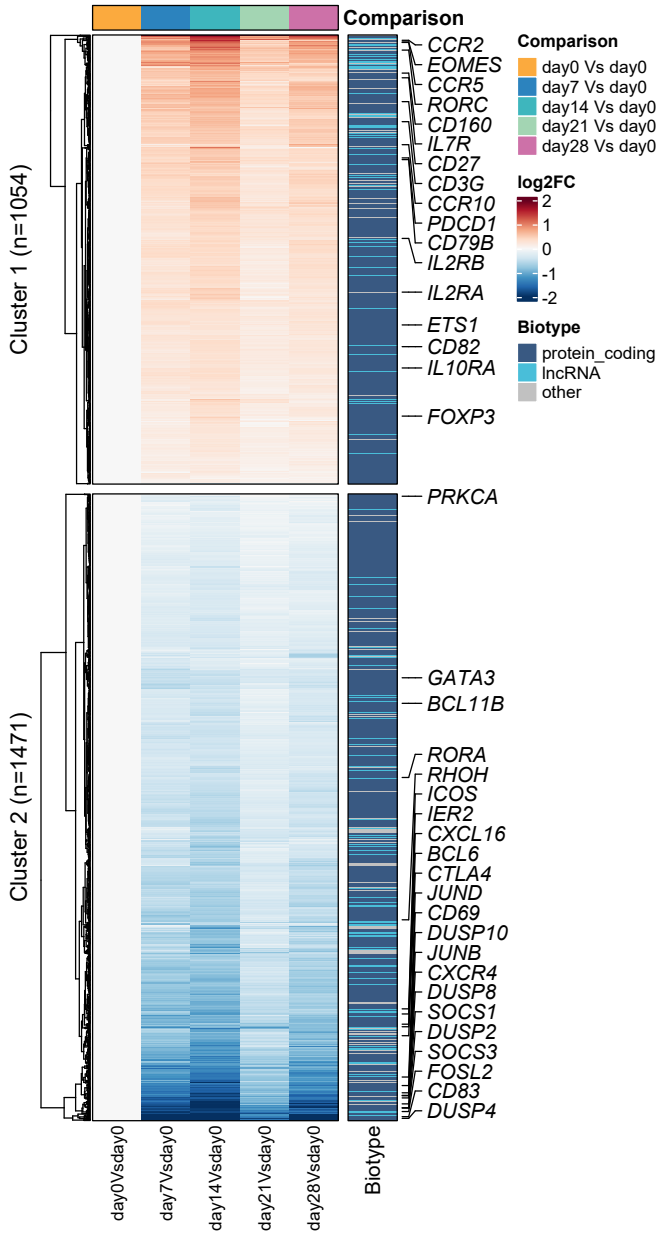

```
ggsave(filename = "plots/fig1G_DEGs_cluster_expr.pdf", width = 5.5, height = 3, units = "in", dpi = 300, device = cairo_pdf)
```

Heatmap with selected genes

```
comparisons <- names(DESeq2_DEGs)[2:length(DESeq2_DEGs)]

DEGs_clusters_FC <- lapply(setNames(comparisons,comparisons), function(x) {
  DESeq2_DEGs[[x]] %>%
    filter(GeneSymbol %in% DEGs$GeneSymbol) %>%
    select(GeneSymbol, log2FoldChange) %>%
    rename_with(~x, log2FoldChange)
}) %>%
  purrr::reduce(left_join, by = 'GeneSymbol') %>%
  add_column(Cluster = recode(.$GeneSymbol, !!!DEGs_clust_k2$cluster), .after = 'GeneSymbol') %>%
  mutate(Cluster = recode(Cluster, '1'='2', '2'='1')) %>%
  add_column(day0Vsday0 = 0, .after = 'Cluster') %>%
  add_column(Geneid = recode(.$GeneSymbol, !!!setNames(RNAseq$filt$annotation$Geneid,
                                                       nm = RNAseq$filt$annotation$GeneSymbol)),
             .before = 'GeneSymbol') %>%
  arrange(Cluster) %>%
  mutate(Biotype = recode(.$GeneSymbol, !!!setNames(RNAseq$filt$annotation$gene_biotype,
                                                    nm = RNAseq$filt$annotation$GeneSymbol)))
```

```
# relevant genes to annotate
mark.genes <- c("IL7R","ETS1","GATA3","TCF7","TCF1","BCL11B",
                "SPI1","HES1","BCL11A","TCF12","BCL6","BCL2",
                "IER2",
                "CD27", # activation marker
                "CD3G",
                "CD69", # Early activation marker
                "CCR10","CCR2","CCR5",
                "CD160", # inhibits t cell activation
                "CD79B",
                "CD82","CD83",
                "RORA","RORC",
                "FOXP3",
                "CTLA4",
                "PDCD1",
                "CXCR4",
                "CXCL16",
                "ICOS",
                "IL2RA","IL2RB",
                "IL10RA",
                "EOMES",
                "SOCS1","SOCS3",
                "RHOH",
                "DUSP1","DUSP2","DUSP4","DUSP8","DUSP10",
                "FOS","FOSL2",
                "JUN","JUNB","JUND",
                "STAT5",
                "PRKCA",
                "ATF2"
```

```
)

m <- DEGs_clusters_FC %>%
  column_to_rownames(var = 'GeneSymbol') %>%
  select(day0Vsday0:day28Vsday0)

m2 <- DEGs_clusters_FC %>%
  column_to_rownames(var = 'GeneSymbol') %>%
  select(Biotype) %>%
  mutate(Biotype = ifelse(Biotype %in% c('protein_coding', 'lncRNA'), Biotype, 'other')) %>%
  mutate(Biotype = factor(Biotype, levels = c('protein_coding', 'lncRNA', 'other')))

comparison <- gsub('Vs', ' Vs ', colnames(m))

lbls <- c("1" = paste0("Cluster 1 (n=",sum(DEGs_clust_expr$cluster=='1'),")"),
          "2" = paste0("Cluster 2 (n=",sum(DEGs_clust_expr$cluster=='2'),")"))

ha_top <- HeatmapAnnotation(
  Comparison = factor(comparison, levels = unique(comparison)),
  col = list(
    Comparison = setNames(colPals$time,
                          nm = unique(comparison))
  ),
  annotation_name_gp = gpar(fontface = 'bold'),
  border = T
)

ha_right <- rowAnnotation(
  mark = anno_mark(at=which(rownames(m) %in% mark.genes),
                   labels = rownames(m)[which(rownames(m) %in% mark.genes)],
                   padding = unit(1,"mm"),
                   labels_gp = gpar(fontface = 'italic'))
)

p <- Heatmap(m, name = "log2FC",
             row_split=DEGs_clusters_FC$Cluster, cluster_row_slices = F, cluster_rows = T,
             column_title = NULL, cluster_columns = F,
             col = circlize::colorRamp2(breaks=seq(-2, 2, length.out=21),
                                        colors=colorRampPalette(rev(colPals$RdBu))(21)),
             top_annotation = ha_top,
             width = unit(50, "mm"),
             show_row_names = F, row_title = lbls, show_row_dend = T, row_dend_width=unit(10, "mm"), row_gap = unit(2, "mm"),
             show_column_names = T, column_names_gp = gpar(fontsize = 10), column_gap = unit(2, "mm"),
             border = T) +
  Heatmap(m2, name = "Biotype",
          col = colPals$biotype,
          right_annotation = ha_right,
          width = unit(10, "mm"),
          show_row_names = F,
          border = T)

draw(p, merge_legend = T, align_heatmap_legend = "heatmap_top")
```



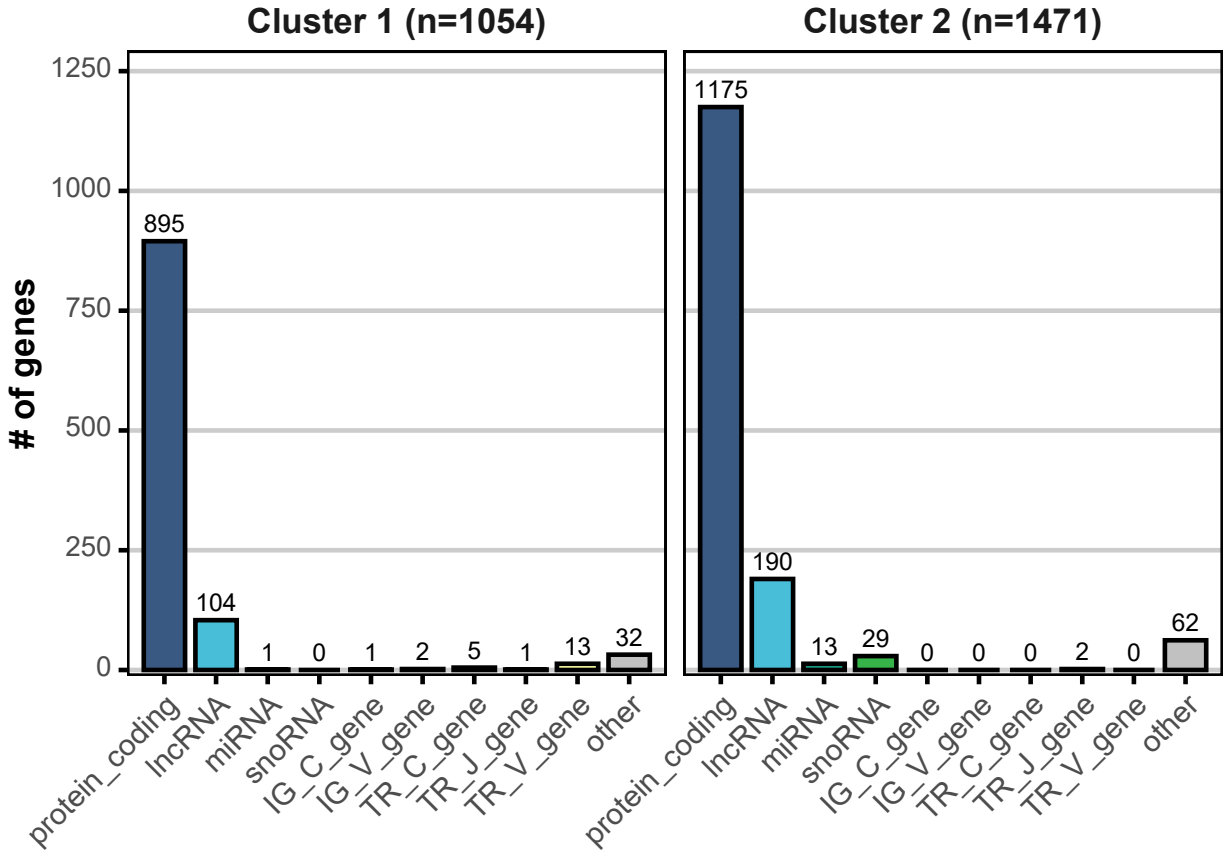

```
pdf("plots/fig2A_DEGs_cluster_heatmap_w_biotype.pdf", width = 10, height = 10)
draw(p, merge_legend = T, align_heatmap_legend = "heatmap_top")
dev.off()
```

```
## cairo_pdf
##         2
```

Biotype distribution

```
df <- DEGs_clusters_FC %>%
  select(Cluster, Biotype) %>%
  mutate(Biotype = ifelse(Biotype %in% names(colPals$biotype), Biotype, 'other')) %>%
  mutate(Biotype = factor(Biotype, levels = names(colPals$biotype))) %>%
  group_by(Cluster,Biotype,.drop = FALSE) %>%
  summarise(n = n())

lbls <- c("1" = paste0("Cluster 1 (n=",sum(DEGs_clust_expr$cluster=='1'),")"),
          "2" = paste0("Cluster 2 (n=",sum(DEGs_clust_expr$cluster=='2'),")"))

ggplot(df, aes(x=Biotype, y=n, fill=Biotype)) +
```

```
  geom_bar(stat = "identity", width = 0.8, size = 1, colour = "black") +
  geom_text(aes(label = n), vjust = -0.5, size = 4) +
  facet_wrap(~Cluster, nrow = 1, labeller = as_labeller(lbls)) +
  xlab("") +
  ylab("# of genes") +
  scale_x_discrete(expand =expansion(mult = c(.08, .08))) +
  scale_y_continuous(expand =expansion(mult = c(.01, .1))) +
  scale_fill_manual(values = colPals$biotype) +
  theme_bw(base_size = 16) +
  theme(
    axis.title.y = element_text(face = 'bold', size = 16),
    axis.text.x.bottom = element_text(angle = 45, hjust = 1, vjust = 1, size = 14),
    axis.text.y = element_text(vjust = 0.3),
    panel.grid.major.y = element_line(color = "grey80", linetype = "solid", size = 1),
    panel.grid.major.x = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    axis.ticks = element_line(color = "black", size = 1),
    legend.position = 'none',
    strip.background = element_blank(),
    strip.text = element_text(face = 'bold', size = 16)
  )
```




```
ggsave(filename = "plots/figS7_DEGs_cluster_biotype_distr.pdf", width = 8, height = 6, units = "in", dpi = 300, device = cairo_pdf)
```

Exports

```
DEGs_clusters <- list(DEGs_clust_expr = DEGs_clust_expr,
                      DEGs_clust_expr_mean = DEGs_clust_expr_mean,
                      DEGs_clusters_FC = DEGs_clusters_FC)
saveRDS(DEGs_clusters, file= "data/rnaseq/DEGs_kmeans_clusters.rds")
```

```
openxlsx::write.xlsx(
  list(DEGs_clusters_log2FC = DEGs_clusters_FC),
  file = "tables/dataS4_DEGs_cluster_assignment.xlsx",
  rowNames=F,
  overwrite=T
)
```

SessionInfo

```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] ComplexHeatmap_2.14.0 cluster_2.1.3         vegan_2.6-4
##  [4] lattice_0.20-45       permute_0.9-7         RColorBrewer_1.1-3
##  [7] patchwork_1.1.2       magrittr_2.0.3        forcats_1.0.0
## [10] stringr_1.5.0         dplyr_1.1.1           purrr_1.0.1
## [13] readr_2.1.4           tidyr_1.3.0           tibble_3.2.1
## [16] ggplot2_3.4.2         tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-157        matrixStats_0.63.0  fs_1.6.1
##  [4] lubridate_1.9.2     doParallel_1.0.17   httr_1.4.5
##  [7] tools_4.2.1         backports_1.4.1     utf8_1.2.3
## [10] R6_2.5.1            DBI_1.1.3           BiocGenerics_0.44.0
## [13] mgcv_1.8-40         colorspace_2.1-0    GetoptLong_1.0.5
## [16] withr_2.5.0         tidyselect_1.2.0    compiler_4.2.1
## [19] cli_3.6.1           rvest_1.0.3         xml2_1.3.3
## [22] labeling_0.4.2      scales_1.2.1        digest_0.6.31
## [25] rmarkdown_2.21      pkgconfig_2.0.3     htmltools_0.5.5
## [28] highr_0.10          dbplyr_2.3.2        fastmap_1.1.1
## [31] rlang_1.1.0         GlobalOptions_0.1.2 readxl_1.4.2
## [34] rstudioapi_0.14     shape_1.4.6         generics_0.1.3
## [37] farver_2.1.1        jsonlite_1.8.4      zip_2.2.2
## [40] googlesheets4_1.1.0 Matrix_1.5-3        Rcpp_1.0.10
## [43] munsell_0.5.0       S4Vectors_0.36.2    fansi_1.0.4
## [46] lifecycle_1.0.3     stringi_1.7.12      yaml_2.3.7
## [49] MASS_7.3-57         parallel_4.2.1      crayon_1.5.2
## [52] haven_2.5.2         splines_4.2.1       circlize_0.4.15
## [55] hms_1.1.3           knitr_1.42          pillar_1.9.0
## [58] rjson_0.2.21        codetools_0.2-18    stats4_4.2.1
## [61] reprex_2.0.2        glue_1.6.2          evaluate_0.20
## [64] modelr_0.1.11       png_0.1-8           vctrs_0.6.1
## [67] tzdb_0.3.0          foreach_1.5.2       cellranger_1.1.0
## [70] gtable_0.3.3        clue_0.3-64         openxlsx_4.2.5.1
## [73] xfun_0.38           broom_1.0.4         googledrive_2.1.0
## [76] gargle_1.3.0        iterators_1.0.14    IRanges_2.32.0
## [79] timechange_0.2.0
```