Step 2.3: RT-qPCR/RNAseq expression measurments for selected genes

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(openxlsx)
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"))
colPals$rtqpcr_rnaseq <- setNames(c("#B80D48","#2B6A6C"),
                                  nm = c("rtqpcr", "rnaseq"))
```

Note:

> Genes for RT-qPCR analysis were selected based on prior interest before RNAseq analysis and are, thus, not entirely driven by it. Housekeeping gene used is HPRT1. HPRT1 has been identified as "stable" gene during simulated microgravity conditions (see Elgindi et al. 2021).

Load data

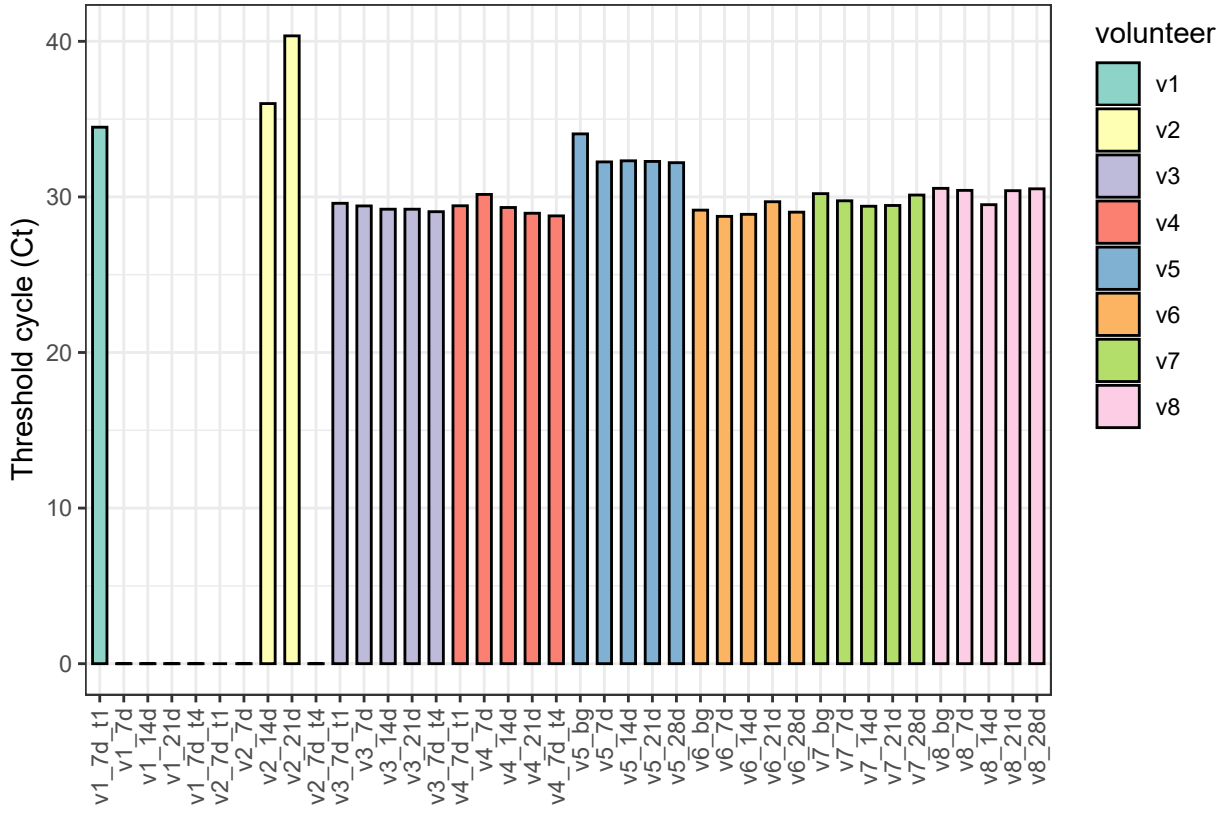
```
# RNA-seq
RNAseq <- readRDS(file='data/rnaseq/rnaseq_volunteers_9&10_excl.rds')
DESeq2_DEGs <- readRDS(file='data/rnaseq/DESeq2_DEGs_unfilt_volunteers_9&10_excl.rds')
DESeq2_DEGs_filt <- readRDS(file='data/rnaseq/DESeq2_DEGs_filt_volunteers_9&10_excl.rds')

# RT-qPCR
RTqPCR_ct <- openxlsx::read.xlsx('data/rtqpcr/rtqpcr_cq_values.xlsx',
                                 startRow = 2,
                                 colNames = T,
                                 rowNames = F) %>%
```

```
  dplyr::rename(sample = Sample.ID) %>%
  filter(!grepl('v\\.(9|10)', sample)) %>%
  select_if(~ !any(is.na(.))) %>%
  mutate(sample = RNAseq$filt$design$sample) %>%
  left_join(RNAseq$filt$design)
```

Housekeeping genes

Cq measurements for V1 and V2 showed high variability indicating experimental difficulties with samples from these volunteers. Also note that samples V9 and V10 are removed according to our exclusion criteria. Overall, RT-qPCR results show high variability with inconsistent trend for some genes between volunteers. However, we identify genes that show a robust profile in all volunteers.

```
ggplot(RTqPCR_ct, aes(x=sample, y=HPRT1, fill=volunteer)) +
  geom_bar(stat='identity', width=0.6, color='black') +
  scale_fill_manual(values = brewer.pal(11, name = "Set3")) +
  xlab('') +
  ylab('Threshold cycle (Ct)') +
  theme_bw() +
  theme(
    axis.text.x.bottom = element_text(angle=90, hjust = 1, vjust = 0.5),
    legend.justification = 'top'
  )
```



Normalize expression

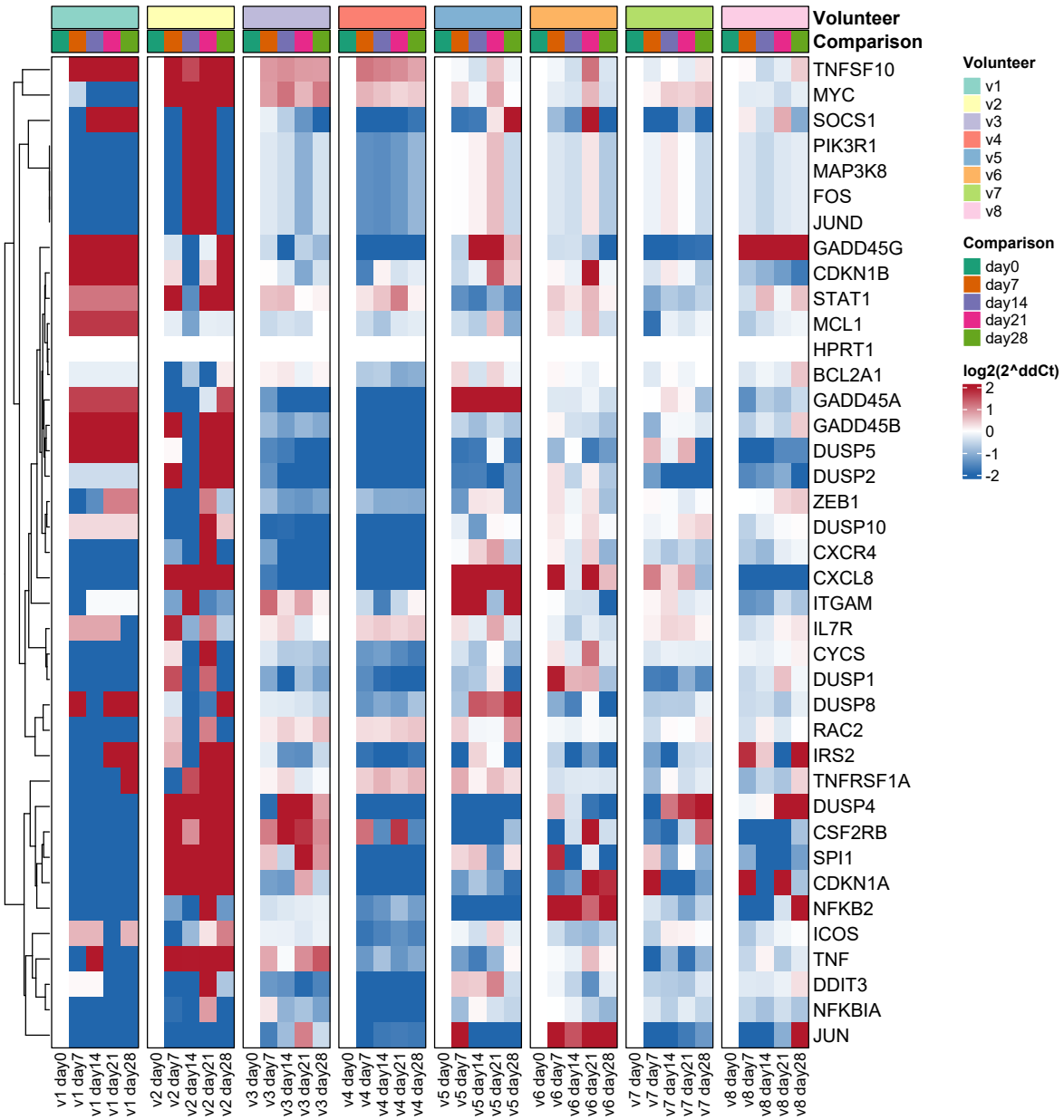
```
# normalize to HPRT1
RTqPCR_dct <- RTqPCR_ct %>%
  column_to_rownames(var = 'sample') %>%
  select(-batch,-volunteer,-time) %>%
```

```
  apply(., MARGIN = 2, FUN=function(x) x - .$HPRT1) %>%
  as.data.frame()

RTqPCR_ddct <- RTqPCR_dct %>%
  mutate(volunteer = RTqPCR_ct$volunteer,
         time = RTqPCR_ct$time) %>%
  group_by(volunteer) %>%
  group_split()

# calculate ddct to day0
RTqPCR_ddct <- lapply(RTqPCR_ddct, function(df) {

  volunteer <- df %>% dplyr::pull(volunteer)
  time <- df %>% dplyr::pull(time)

  baseline <- df %>%
    filter(time == 'day0') %>%
    select(-volunteer, -time) %>%
    as.numeric()

  df %>%
    select(-volunteer, -time) %>%
    apply(., MARGIN = 1, FUN=function(x) x - baseline) %>%
    t() %>%
    as.data.frame() %>%
    mutate(volunteer = volunteer,
           time = time)

}) %>% bind_rows()

# calculate expression as 2^-ddCT
RTqPCR_expr <- RTqPCR_ddct
RTqPCR_expr[,1:39] <- 2^(-RTqPCR_expr[,1:39])

# calculate log2 expression
RTqPCR_log2expr <- RTqPCR_expr
RTqPCR_log2expr[,1:39] <- log2(RTqPCR_log2expr[,1:39])
```

Heatmap with quantifications relative to day0

```
m <- t(RTqPCR_log2expr[,1:39])
colnames(m) <- paste(RTqPCR_log2expr$volunteer, RTqPCR_log2expr$time)

volunteer <- gsub('(v\\d+) .*$', '\\1', colnames(m))
comparison <- gsub('_', ' Vs ', gsub('v\\d+ ', '', colnames(m)))

ha_top <- HeatmapAnnotation(
  Volunteer = factor(volunteer, levels = unique(volunteer)),
  Comparison = factor(comparison, levels = unique(comparison)),
  col = list(
    Volunteer = setNames(brewer.pal(length(unique(volunteer)),"Set3"),
                         nm = unique(volunteer)),
    Comparison = setNames(brewer.pal(length(unique(comparison)),"Dark2"),
                         nm = unique(comparison))
  ),
  annotation_name_gp = gpar(fontface = 'bold'),
  border = T
)

p <- Heatmap(m, name = "log2(2^ddCt)",
             cluster_row_slices = F,
             cluster_rows = T,
             column_title = NULL,
             column_split=factor(volunteer, levels = unique(volunteer)),
             cluster_columns = F,
             col = circlize::colorRamp2(breaks=seq(-2, 2, length.out=21),
                                        colors=colorRampPalette(c("#2166AC", "white", "#B2182B"))(21)),
             top_annotation = ha_top,
             width = unit(170, "mm"),
             show_row_names = T,
             row_title = NULL,
             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)

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



Compare RT-qPCR and RNAseq

```
# RT-qPCR genes (without housekeeping: HPRT1)
RTqPCR_genes <- colnames(RTqPCR_log2expr[,2:39])

# calculate fold changes per volunteer
RNAseq_log2FC <- calcPerVolunteerFCs(log2(RNAseq$filt$DESeq_norm),
                                     RNAseq$filt$design,
                                     comparison = list(
                                       c("day0","day0"),
                                       c("day7","day0"),
                                       c("day14","day0"),
                                       c("day21","day0"),
```

```
                                    c("day28","day0")
                                  ))
```

```
## [1] "v1" "v2" "v3" "v4" "v5" "v6" "v7" "v8"
## [1] "v1"
## [1] "v2"
## [1] "v3"
## [1] "v4"
## [1] "v5"
## [1] "v6"
## [1] "v7"
## [1] "v8"
```

```
RNAseq_log2FC <- RNAseq_log2FC[!apply(RNAseq_log2FC,1,FUN=function(x) any(is.infinite(x))),]
RNAseq_log2FC[is.na(RNAseq_log2FC)] <- 0
RNAseq_log2FC <- RNAseq_log2FC %>%
  as.data.frame() %>%
  rownames_to_column(var = 'GeneSymbol') %>%
  mutate(GeneSymbol = gsub('(.+)_\\d+','\\1',GeneSymbol)) %>%
  filter(GeneSymbol %in% RTqPCR_genes) %>%
  filter(!duplicated(GeneSymbol))

# shared genes
shared_genes <- RNAseq_log2FC$GeneSymbol

# long format data frames
RTqPCR_log2expr_long <- RTqPCR_log2expr %>%
  select(shared_genes, 'volunteer', 'time') %>%
  pivot_longer(shared_genes, names_to = 'GeneSymbol', values_to = 'log2FC') %>%
  mutate(type = 'rtqpcr',
         volunteer = as.character(volunteer),
         time = as.character(time))

RNAseq_log2FC_long <- RNAseq_log2FC %>%
  column_to_rownames(var = 'GeneSymbol') %>%
  t() %>%
  as.data.frame() %>%
  mutate(volunteer = gsub('(v\\d+) .*','\\1',rownames(.)),
         time = gsub('v\\d+ (day\\d+)_day0','\\1',rownames(.))) %>%
  select(shared_genes, 'volunteer', 'time') %>%
  pivot_longer(shared_genes, names_to = 'GeneSymbol', values_to = 'log2FC') %>%
  mutate(type = 'rnaseq')

# long format gene mean expression per time point
RTqPCR_log2expr_long_mean <- RTqPCR_log2expr_long %>%
  filter(!volunteer %in% c('v1','v2')) %>%
  group_by(GeneSymbol, time) %>%
  summarise(log2FC = mean(log2FC)) %>%
  arrange(factor(GeneSymbol, levels = shared_genes),
          factor(time, levels = names(colPals$time))) %>%
  mutate(type = 'rtqpcr')

RNAseq_log2FC_long_mean <- RNAseq_log2FC_long %>%
  filter(!volunteer %in% c('v1','v2')) %>%
  group_by(GeneSymbol, time) %>%
  summarise(log2FC = mean(log2FC)) %>%
  arrange(factor(GeneSymbol, levels = shared_genes),
          factor(time, levels = names(colPals$time))) %>%
  mutate(type = 'rnaseq')
```

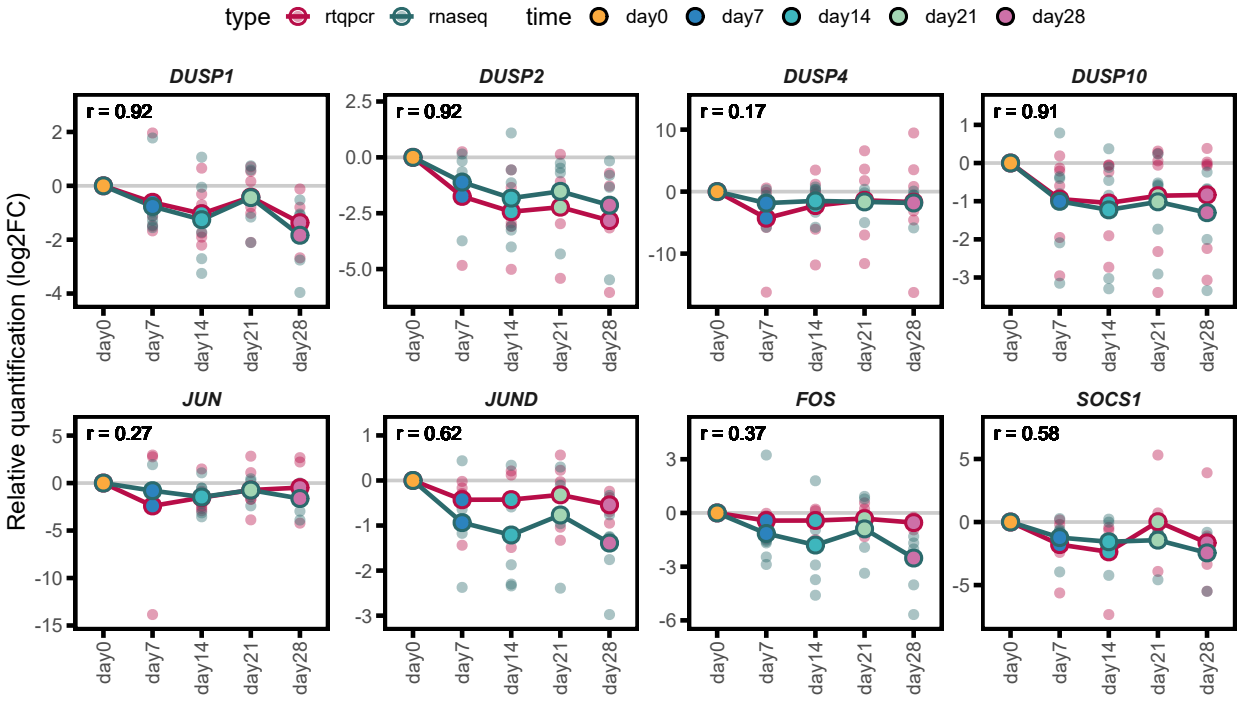
Correlation

```
# we exclude V1 and V2 due to high variability and inflated expression measurements
volunteer_order <- paste('v', seq(3,8), sep = '')
time_order <- paste('day', c(0,7,14,21,28), sep = '')

# calculate correlations
res_cor <- lapply(setNames(shared_genes, shared_genes), function(x) {

  rtqpcr_log2FC <- RTqPCR_log2expr_long %>%
    filter(volunteer %in% volunteer_order & GeneSymbol == x) %>%
    arrange(factor(volunteer, levels = volunteer_order),
            factor(time, levels = time_order)) %>%
    pull(log2FC)

  rnaseq_log2FC <- RNAseq_log2FC_long %>%
    filter(volunteer %in% volunteer_order & GeneSymbol == x) %>%
    arrange(factor(volunteer, levels = volunteer_order),
            factor(time, levels = time_order)) %>%
```

```
    pull(log2FC)

  rtqpcr_rnaseq_cor <- cor(rtqpcr_log2FC, rnaseq_log2FC, method = "pearson")

  data.frame(GeneSymbol = x,
             r = rtqpcr_rnaseq_cor)

}) %>% bind_rows()


# calculate correlations per volunteer
gene_volunteer_comb <- expand.grid(list(GeneSymbol = shared_genes,
                                        volunteer = unique(volunteer))) %>%
  filter(!duplicated(.))
gene_volunteer_comb <- split(gene_volunteer_comb, seq(nrow(gene_volunteer_comb)))

res_cor_by_volunteer <- lapply(gene_volunteer_comb, function(x) {

  rtqpcr_log2FC <- RTqPCR_log2expr_long %>%
    filter(volunteer == x$volunteer & GeneSymbol == x$GeneSymbol) %>%
    arrange(factor(time, levels = time_order)) %>%
    pull(log2FC)

  rnaseq_log2FC <- RNAseq_log2FC_long %>%
    filter(volunteer == x$volunteer & GeneSymbol == x$GeneSymbol) %>%
    arrange(factor(time, levels = time_order)) %>%
    pull(log2FC)

  rtqpcr_rnaseq_cor <- cor(rtqpcr_log2FC, rnaseq_log2FC, method = "pearson")

  data.frame(volunteer = x$volunteer,
             GeneSymbol = x$GeneSymbol,
             r = rtqpcr_rnaseq_cor)

}) %>% bind_rows()

res_cor_by_volunteer_mean <- res_cor %>%
  group_by(GeneSymbol) %>%
  summarise(r_mean = mean(r))
```

```
selected_genes <- c('DUSP1','DUSP2','DUSP4','DUSP10','JUN','JUND','FOS','SOCS1')

df <- bind_rows(RTqPCR_log2expr_long,
                RNAseq_log2FC_long) %>%
  filter(!volunteer %in% c('v1','v2')) %>%
  mutate(r = recode(GeneSymbol, !!!setNames(res_cor$r,
                                            nm = res_cor$GeneSymbol))) %>%
  filter(GeneSymbol %in% selected_genes) %>%
  mutate(GeneSymbol = factor(GeneSymbol, levels = selected_genes),
         time = factor(time, levels = names(colPals$time)),
         type = factor(type, levels = names(colPals$rtqpcr_rnaseq)))

df2 <- bind_rows(RTqPCR_log2expr_long_mean,
                 RNAseq_log2FC_long_mean) %>%
  mutate(r = recode(GeneSymbol, !!!setNames(res_cor$r,
                                            nm = res_cor$GeneSymbol))) %>%
  filter(GeneSymbol %in% selected_genes) %>%
  mutate(GeneSymbol = factor(GeneSymbol, levels = selected_genes),
         time = factor(time, levels = names(colPals$time)),
         type = factor(type, levels = names(colPals$rtqpcr_rnaseq)))

ggplot(df, aes(x=time, y=log2FC, color=type)) +
  geom_hline(yintercept = 0, color='grey80', size=1) +
  geom_point(shape=16, size=3, stroke=0, alpha=0.4) +
  facet_wrap(~GeneSymbol, ncol=4, scales = 'free') +
  geom_line(data = df2, aes(x=time, y=log2FC, group=type, color=type), size = 1.25) +
  geom_point(data = df2, aes(x=time, y=log2FC, group=type, fill=time), shape=21, size=3.5, stroke=1.25) +
  geom_text(aes(label=paste('r =',round(r,2)), group=r),x = -Inf, y = Inf, hjust = -0.2, vjust = 1.8, inherit.aes = FALSE) +
  scale_y_continuous(expand = expansion(mult = c(.1, .25))) +
  xlab('') +
  ylab('Relative quantification (log2FC)') +
  scale_color_manual(values = c(colPals$time, colPals$rtqpcr_rnaseq)) +
  scale_fill_manual(values = c(colPals$time, colPals$rtqpcr_rnaseq)) +
  theme_bw() +
  theme(
    text = element_text(family = 'Arial', size = 14),
    axis.text.x.bottom = element_text(angle = 90, hjust = 1, vjust = 0.5),
    panel.border = element_rect(color = "black", fill = NA, size = 2),
    axis.ticks = element_line(color = "black", size = 1.25),
    axis.ticks.length = unit(1.5, 'mm'),
```

```
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank(),
    panel.grid.major.y = element_blank(),
    panel.grid.minor.y = element_blank(),
    legend.position = 'top',
    strip.background = element_blank(),
    strip.text = element_text(face = "bold.italic"),
    strip.text.y = element_text(angle = 90)
  )
```




```
ggsave(filename = "plots/fig2B_RTqPCR_RNAseq_selected_log2FC.pdf", width = 10, height = 6, units = "in", dpi = 300, device = cairo_pdf)
```

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 openxlsx_4.2.5.1      RColorBrewer_1.1-3
##  [4] patchwork_1.1.2       magrittr_2.0.3        forcats_1.0.0
##  [7] stringr_1.5.0         dplyr_1.1.1           purrr_1.0.1
## [10] readr_2.1.4           tidyr_1.3.0           tibble_3.2.1
## [13] ggplot2_3.4.2         tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
```

```
##  [1] httr_1.4.5          jsonlite_1.8.4      foreach_1.5.2
##  [4] modelr_0.1.11       highr_0.10          stats4_4.2.1
##  [7] googlesheets4_1.1.0 cellranger_1.1.0    yaml_2.3.7
## [10] pillar_1.9.0        backports_1.4.1     glue_1.6.2
## [13] digest_0.6.31       rvest_1.0.3         colorspace_2.1-0
## [16] htmltools_0.5.5     pkgconfig_2.0.3     GetoptLong_1.0.5
## [19] broom_1.0.4         haven_2.5.2         scales_1.2.1
## [22] tzdb_0.3.0          timechange_0.2.0    googledrive_2.1.0
## [25] farver_2.1.1        generics_0.1.3      IRanges_2.32.0
## [28] withr_2.5.0         BiocGenerics_0.44.0 cli_3.6.1
## [31] crayon_1.5.2        readxl_1.4.2        evaluate_0.20
## [34] fs_1.6.1            fansi_1.0.4         doParallel_1.0.17
## [37] xml2_1.3.3          tools_4.2.1         hms_1.1.3
## [40] GlobalOptions_0.1.2 gargle_1.3.0        lifecycle_1.0.3
## [43] matrixStats_0.63.0  S4Vectors_0.36.2    munsell_0.5.0
## [46] reprex_2.0.2        cluster_2.1.3       zip_2.2.2
## [49] compiler_4.2.1      rlang_1.1.0         iterators_1.0.14
## [52] rstudioapi_0.14     circlize_0.4.15     rjson_0.2.21
## [55] labeling_0.4.2      rmarkdown_2.21      gtable_0.3.3
## [58] codetools_0.2-18    DBI_1.1.3           R6_2.5.1
## [61] lubridate_1.9.2     knitr_1.42          fastmap_1.1.1
## [64] utf8_1.2.3          clue_0.3-64         shape_1.4.6
## [67] stringi_1.7.12      parallel_4.2.1      Rcpp_1.0.10
## [70] vctrs_0.6.1         png_0.1-8           dbplyr_2.3.2
## [73] tidyselect_1.2.0    xfun_0.38
```