Step 3.2: FACS analysis

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)

# 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"))
```

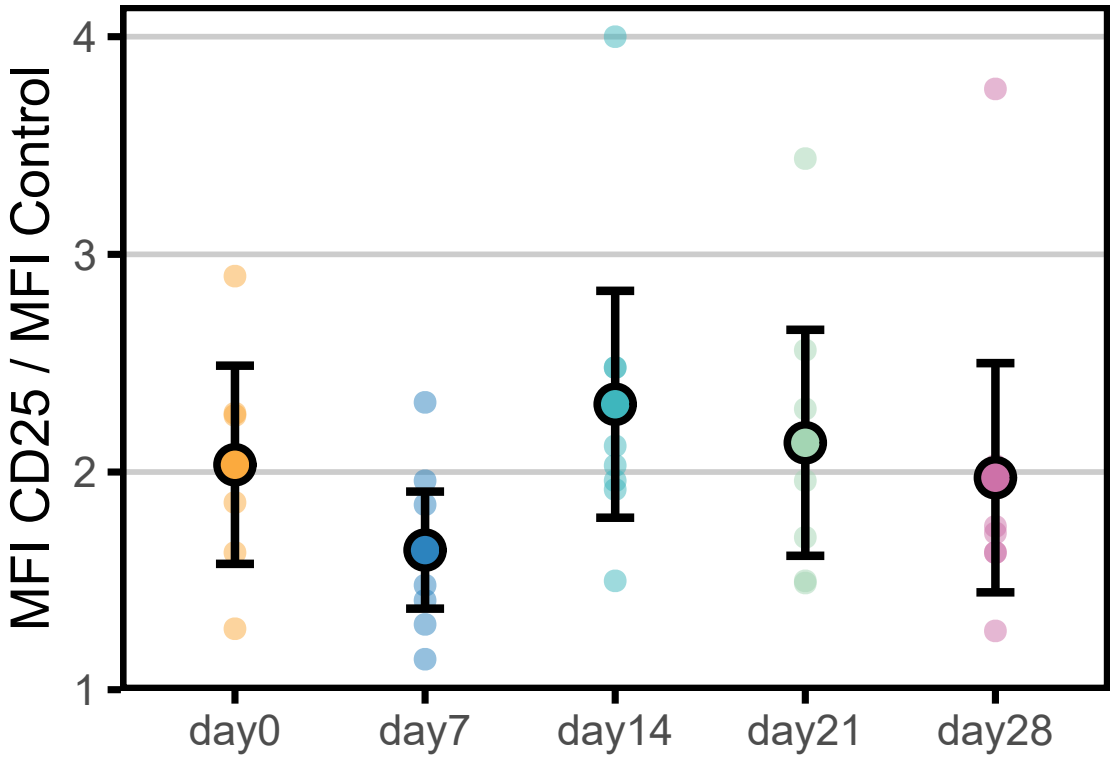
Load data

```
# FACS data
filenames <- list.files(path= "data/facs", pattern = "^facs_",full.names = T)

facs_data <- lapply(filenames, function(x) {

  valunteer <- gsub('.*(v\\d+).*', '\\1', x)

  df <- read.table(file = x,
                   stringsAsFactors = F,
                   sep = ",",
                   header = T,
                   fill = T,
                   quote = "")

  colnames(df) <- c('time', 'gate', 'perc_among_CD3', 'perc_among_lymphocytes',
                    'MFI', 'MFI_CD25_peak_vs_MFI_contr_lymph_peak')

  df <- df %>%
    add_column(volunteer = valunteer, .before = 'time') %>%
    mutate(perc_among_CD3 = as.numeric(perc_among_CD3),
           perc_among_lymphocytes = as.numeric(perc_among_lymphocytes),
           MFI = as.numeric(MFI),
           MFI_CD25_peak_vs_MFI_contr_lymph_peak = as.numeric(MFI_CD25_peak_vs_MFI_contr_lymph_peak))
```

```
  df

}) %>% bind_rows()
```

Mean MFI of CD25 over control

```
# mean fluorescence intensity (MFI) at different time points
df <- facs_data %>%
  filter(volunteer %in% paste0('v', seq(1,8))) %>%
  filter(!is.na(MFI_CD25_peak_vs_MFI_contr_lymph_peak)) %>%
  filter(MFI_CD25_peak_vs_MFI_contr_lymph_peak != 0) %>%
  mutate(time = recode(time, !!!setNames(names(colPals$time),
                                         nm = c('-7d','7d','14d','21d','+7d')))) %>%
  filter(time %in% names(colPals$time)) %>%
  mutate(time = factor(time, levels = names(colPals$time)))

df2 <- df %>%
  group_by(time) %>%
  select(time, MFI_CD25_peak_vs_MFI_contr_lymph_peak) %>%
  summarize_each(dplyr::funs(mean, sd, se=sd(.)/sqrt(n())), MFI_CD25_peak_vs_MFI_contr_lymph_peak)

ggplot() +
  geom_point(data = df, aes(x=time, y=MFI_CD25_peak_vs_MFI_contr_lymph_peak, color=time),
             shape=16, size=4, stroke=0, alpha=0.5) +
  geom_errorbar(data=df2, aes(x=time, y=mean, ymin=mean-se*1.96, ymax=mean+se*1.96), width=.2, lwd=1.5) +
  # geom_line(data = df2, aes(x=time, y=mean, group=1), color='black', size = 1.5) +
  geom_point(data = df2, aes(x=time, y=mean, fill=time), color="black", shape=21, size=5, stroke=2, alpha=1) +
  scale_y_continuous(expand = expansion(mult = c(.05, .05))) +
  scale_color_manual(values = colPals$time) +
  scale_fill_manual(values = colPals$time) +
  xlab('') +
  ylab('MFI CD25 / MFI Control') +
  theme_bw(base_size = 20) +
  theme(panel.grid.major.y = element_line(color = "grey80", linetype = "solid", size = 1),
        panel.grid.minor.y = element_blank(),
        panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank(),
        panel.border = element_rect(color = "black", fill = NA, size = 2),
        axis.ticks = element_line(color = "black", size = 1.25),
        legend.position = "none"
  )
```




```
ggsave(filename = "plots/fig3C_FACS_MFI_CD25_over_control.pdf", width = 6, height = 4.5, units = "in", dpi = 300, device = cairo_pdf)
```

```
# perform paired t-tests
tests_comb <- expand.grid(time1 = names(colPals$time), time2 = names(colPals$time)) %>%
  filter(time1 != time2) %>%
  t() %>%
  as.data.frame()

res_t_test <- lapply(tests_comb, function(x) {
  time1 <- x[1]
  time2 <- x[2]

  group1 <- df %>%
    filter(time == time1) %>%
    pull(MFI_CD25_peak_vs_MFI_contr_lymph_peak)

  id1 <- df %>%
    filter(time == time1) %>%
    pull(volunteer)

  group2 <- df %>%
    filter(time == time2) %>%
    pull(MFI_CD25_peak_vs_MFI_contr_lymph_peak)

  id2 <- df %>%
    filter(time == time2) %>%
    pull(volunteer)

  test_table <- data.frame(volunteer = unique(df$volunteer)) %>%
    mutate(group1 = recode(volunteer, !!!setNames(group1,
                                                  id1)),
           group2 = recode(volunteer, !!!setNames(group2,
                                                  id2))) %>%
    mutate(group1 = as.numeric(group1),
           group2 = as.numeric(group2))

  c(time_1 = time1,
    time_2 = time2,
    p_val = t.test(test_table$group1, test_table$group2, paired = T)$p.value)
```

```
}) %>% bind_rows() %>%
  mutate(p_adj = p.adjust(p_val, method = 'BH'))

res_t_test
```

```
## # A tibble: 20 x 4
##    time_1 time_2 p_val              p_adj
##    <chr>  <chr>  <chr>              <dbl>
##  1 day7   day0   0.102276964699089  0.311
##  2 day14  day0   0.182054125417157  0.311
##  3 day21  day0   0.186706873102431  0.311
##  4 day28  day0   0.804372845689488  0.804
##  5 day0   day7   0.102276964699089  0.311
##  6 day14  day7   0.0761883643823477 0.311
##  7 day21  day7   0.163774733496774  0.311
##  8 day28  day7   0.366268787921531  0.458
##  9 day0   day14  0.182054125417157  0.311
## 10 day7   day14  0.0761883643823477 0.311
## 11 day21  day14  0.271483492878872  0.388
## 12 day28  day14  0.0517690280085766 0.311
## 13 day0   day21  0.186706873102431  0.311
## 14 day7   day21  0.163774733496774  0.311
## 15 day14  day21  0.271483492878872  0.388
## 16 day28  day21  0.729468613696315  0.804
## 17 day0   day28  0.804372845689488  0.804
## 18 day7   day28  0.366268787921531  0.458
## 19 day14  day28  0.0517690280085766 0.311
## 20 day21  day28  0.729468613696315  0.804
```

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] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] openxlsx_4.2.5.1   RColorBrewer_1.1-3 patchwork_1.1.2    magrittr_2.0.3
##  [5] forcats_1.0.0      stringr_1.5.0      dplyr_1.1.1        purrr_1.0.1
##  [9] readr_2.1.4        tidyr_1.3.0        tibble_3.2.1       ggplot2_3.4.2
## [13] tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.2.0    xfun_0.38           haven_2.5.2
##  [4] gargle_1.3.0        colorspace_2.1-0    vctrs_0.6.1
##  [7] generics_0.1.3      htmltools_0.5.5     yaml_2.3.7
## [10] utf8_1.2.3          rlang_1.1.0         pillar_1.9.0
## [13] glue_1.6.2          withr_2.5.0         DBI_1.1.3
## [16] dbplyr_2.3.2        modelr_0.1.11       readxl_1.4.2
## [19] lifecycle_1.0.3     munsell_0.5.0       gtable_0.3.3
## [22] cellranger_1.1.0    zip_2.2.2           rvest_1.0.3
## [25] evaluate_0.20       labeling_0.4.2      knitr_1.42
## [28] tzdb_0.3.0          fastmap_1.1.1       fansi_1.0.4
## [31] highr_0.10          Rcpp_1.0.10         broom_1.0.4
## [34] scales_1.2.1        backports_1.4.1     googlesheets4_1.1.0
## [37] jsonlite_1.8.4      farver_2.1.1        fs_1.6.1
## [40] hms_1.1.3           digest_0.6.31       stringi_1.7.12
## [43] grid_4.2.1          cli_3.6.1           tools_4.2.1
## [46] crayon_1.5.2        pkgconfig_2.0.3     xml2_1.3.3
## [49] reprex_2.0.2        googledrive_2.1.0   lubridate_1.9.2
## [52] timechange_0.2.0    rmarkdown_2.21      httr_1.4.5
## [55] rstudioapi_0.14     R6_2.5.1            compiler_4.2.1
```