Step 1.2: Cell type deconvolution

Carlos Gallardo & Christian Oertlin

17 April, 2023

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

# 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"))
```

Load data

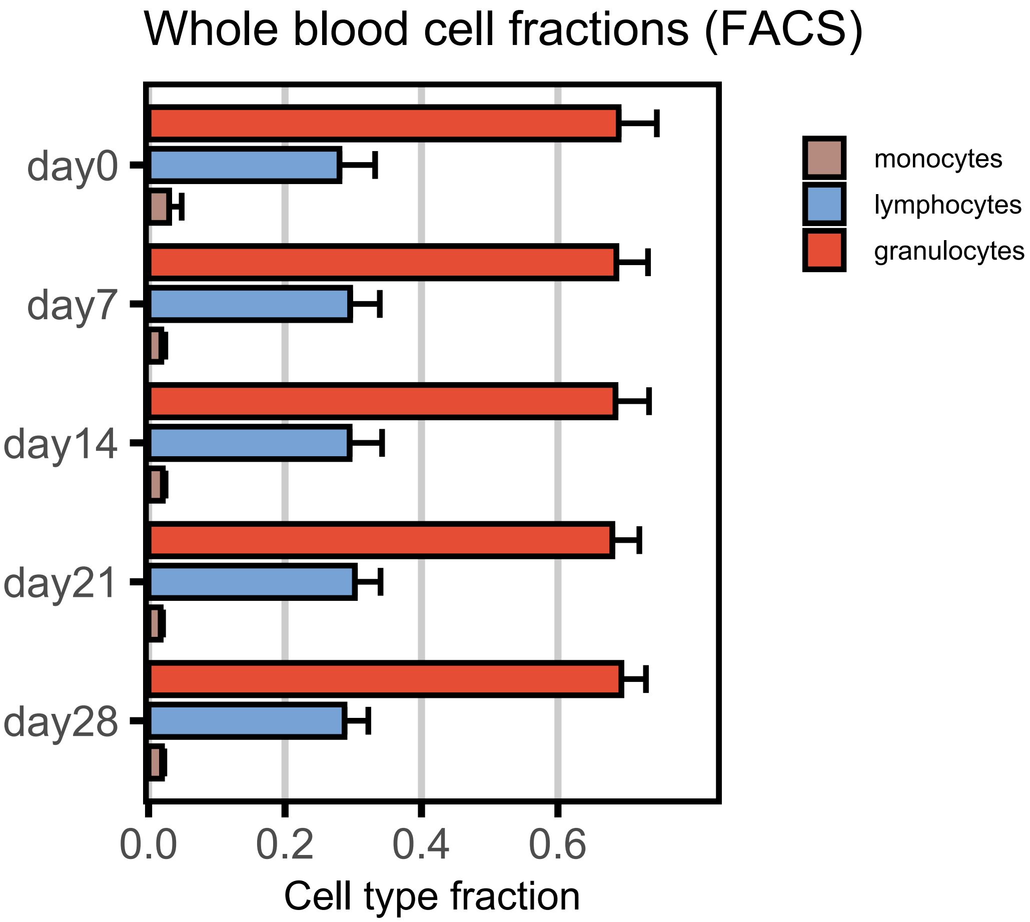
```
# RNA-seq
RNAseq <- readRDS(file='data/rnaseq/rnaseq_volunteers_9&10_excl.rds')
DESeq2_DEGs <- readRDS(file='data/rnaseq/DESeq2_DEGs_unfilt_volunteers_9&10_excl.rds')
DESeq2_DEGs_filt <- readRDS(file='data/rnaseq/DESeq2_DEGs_filt_volunteers_9&10_excl.rds')

# FACS
facs_frac <- read.table(file = 'data/facs/cellType_percentages_facs.csv',
                        stringsAsFactors = F,
                        sep = ",",
                        header = T)
```

Cell type fractions (FACS)

```
df <- facs_frac %>%
  pivot_longer(day0:day28, names_to = 'time', values_to = 'fraction') %>%
  filter(!(is.na(fraction))) %>%
  mutate(fraction = fraction/100,
         type = factor(type, levels = c('monocytes','lymphocytes','granulocytes')),
         time = factor(time, levels = names(colPals$time))) %>%
  group_by(time, type) %>%
  summarize_each(dplyr::funs(mean,
                             sd,
                             se=sd(.)/sqrt(n())), fraction)

df
```

```
## # A tibble: 15 x 5
## # Groups:   time [5]
##    time  type           mean      sd      se
##    <fct> <fct>         <dbl>   <dbl>   <dbl>
##  1 day0  monocytes    0.0302 0.0296  0.00935
##  2 day0  lymphocytes  0.280  0.0843  0.0267
##  3 day0  granulocytes 0.689  0.0903  0.0286
##  4 day7  monocytes    0.0188 0.00840 0.00297
##  5 day7  lymphocytes  0.296  0.0626  0.0221
##  6 day7  granulocytes 0.686  0.0668  0.0236
##  7 day14 monocytes    0.0209 0.00644 0.00204
##  8 day14 lymphocytes  0.295  0.0767  0.0243
##  9 day14 granulocytes 0.684  0.0793  0.0251
## 10 day21 monocytes    0.0178 0.00549 0.00174
## 11 day21 lymphocytes  0.302  0.0607  0.0192
## 12 day21 granulocytes 0.680  0.0638  0.0202
## 13 day28 monocytes    0.0197 0.00552 0.00175
## 14 day28 lymphocytes  0.287  0.0563  0.0178
## 15 day28 granulocytes 0.693  0.0580  0.0183
```

```
ggplot(df, aes(x=time, y=mean, group=type, fill=type)) +
  geom_bar(stat="identity", position = position_dodge(0.9), width = 0.7, size=1, color='black') +
  geom_errorbar(aes(ymin=mean, ymax=mean+se*1.96), width=.6, lwd=1, position = position_dodge(width = 0.9)) +
  scale_fill_manual(values = colPals$blood_cells) +
  scale_x_discrete(limits=rev) +
  scale_y_continuous(limits = c(0,0.8),
                     breaks = seq(0,0.6,0.2),
                     expand = expansion(mult = c(.01, .05))) +
  coord_flip() +
  xlab("") +
  ylab("Cell type fraction") +
  ggtitle("Whole blood cell fractions (FACS)") +
  theme_bw(base_size = 20) +
  theme(
    title = element_text(size=15),
    panel.grid.major.x = element_line(color = "grey80", linetype = "solid", size = 1.25),
    panel.grid.major.y = 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.title = element_blank(),
    legend.justification = 'top',
    legend.text = element_text(size=10)
  )
```



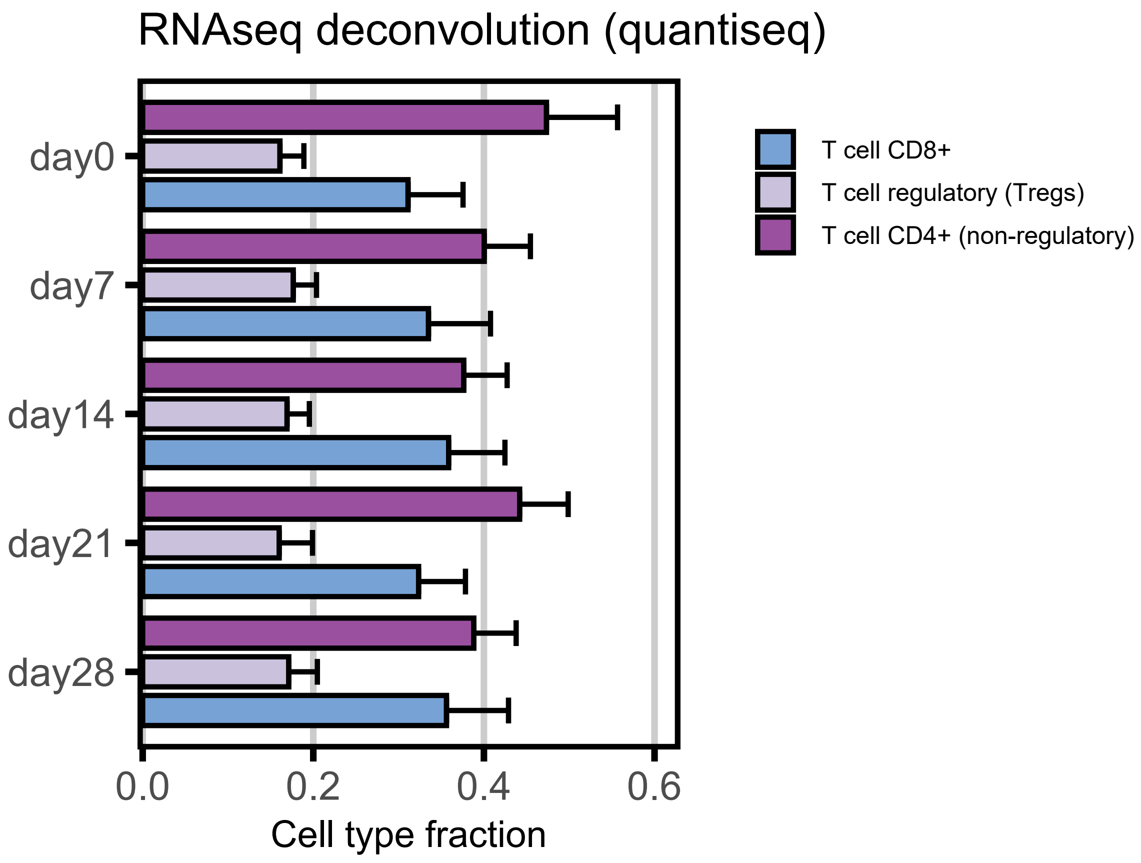
```
ggsave(filename = "plots/fig1B_whole_blood_cell_fraction.pdf", width = 6, height = 5, units = "in", dpi = 300, device = cairo_pdf)
```

Cell type deconvolution (quantiseq)

```
# Run deconvolution
df <- RNAseq$filt$cpm %>%
  rownames_to_column(var = 'GeneSymbol') %>%
  mutate(GeneSymbol = gsub('(.+)_\\d+', '\\1', GeneSymbol)) %>%
  filter(!duplicated(GeneSymbol)) %>%
  column_to_rownames(var = 'GeneSymbol')
rnaseq_deconv <- immunedeconv::deconvolute(df, method = "quantiseq")

df <- rnaseq_deconv %>%
  column_to_rownames(var = 'cell_type') %>%
  t() %>%
  as.data.frame() %>%
  rownames_to_column(var = 'sample') %>%
  left_join(RNAseq$filt$design, by = 'sample') %>%
  pivot_longer(`B cell`:`uncharacterized cell`, names_to = 'cell_type', values_to = 'fraction')
```

Without non-T cell types

```
df2 <- df %>%
  filter(cell_type %in% c('T cell CD4+ (non-regulatory)',
                          'T cell CD8+',
                          'T cell regulatory (Tregs)')) %>%
  mutate(cell_type = factor(cell_type,
                            levels = c('T cell CD8+',
                                       'T cell regulatory (Tregs)',
                                       'T cell CD4+ (non-regulatory)'))) %>%
  group_by(time, cell_type) %>%
  summarize_each(dplyr::funs(mean, sd, se=sd(.)/sqrt(n())), fraction)

ggplot(df2, aes(x=time, y=mean, group=cell_type, fill=cell_type)) +
  geom_bar(stat="identity", position = position_dodge(0.9), width = 0.7, size=1, color='black') +
  geom_errorbar(aes(ymin=mean, ymax=mean+se*1.96), width=.6, lwd=1, position = position_dodge(width = 0.9)) +
  scale_fill_manual(values = colPals$cell_types) +
  scale_x_discrete(limits=rev) +
  scale_y_continuous(limits = c(0,0.6),
                     breaks = seq(0,0.6,0.2),
                     expand = expansion(mult = c(.01, .05))) +
  coord_flip() +
  xlab("") +
  ylab("Cell type fraction") +
  ggtitle("RNAseq deconvolution (quantiseq)") +
  theme_bw(base_size = 20) +
  theme(
    title = element_text(size=15),
    panel.grid.major.x = element_line(color = "grey80", linetype = "solid", size = 1.25),
    panel.grid.major.y = 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.title = element_blank(),
    legend.justification = 'top',
    legend.text = element_text(size=10)
  )
```



```
ggsave(filename = "plots/fig1C_cell_type_deconvolution.pdf", width = 7, height = 5, units = "in", dpi = 300, device = cairo_pdf)
```

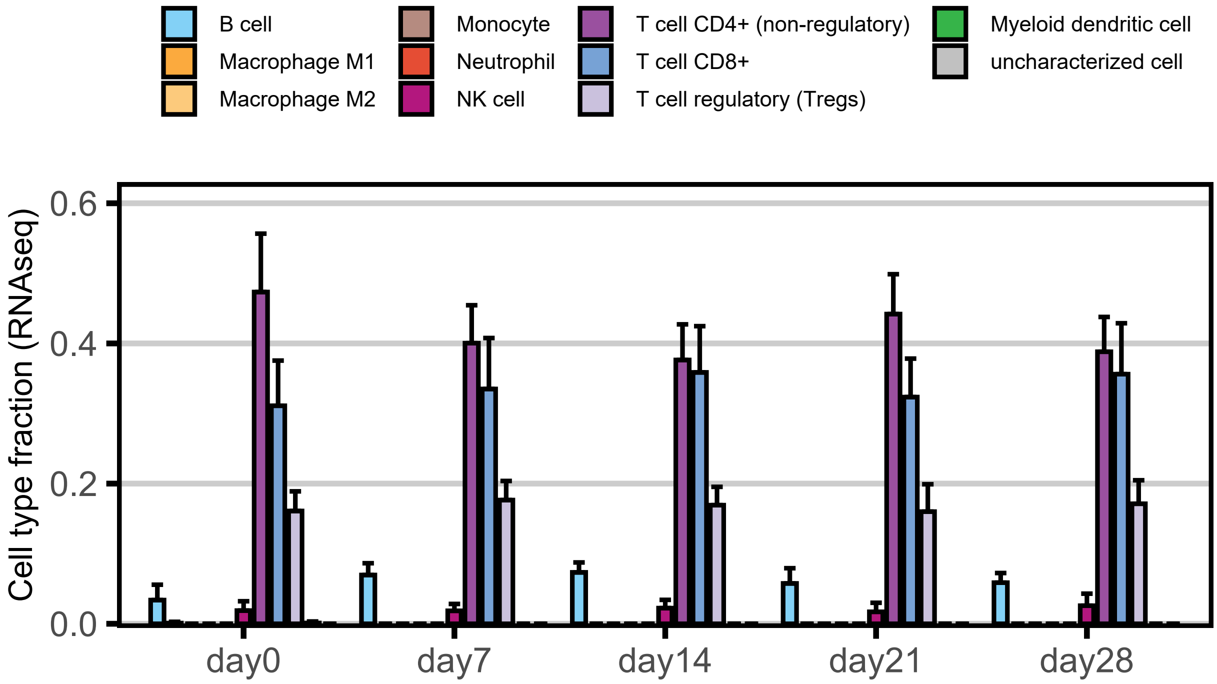
```
# Perform paired t-tests
T_cell_types <- c('T cell CD4+ (non-regulatory)', 'T cell regulatory (Tregs)', 'T cell CD8+')
comparisons <- c('day7_day0','day14_day0','day21_day0','day28_day0')
tests_comb <- expand.grid(cell_type = T_cell_types, time_comp = comparisons) %>%
  t() %>%
  as.data.frame()
```

```
res_t_test <- lapply(tests_comb, function(x) {
  cell <- x[1]
  time1 <- str_split(x[2], pattern = '_')[[1]][1]
  time2 <- str_split(x[2], pattern = '_')[[1]][2]

  group1 <- df %>%
    filter(cell_type == cell & time == time1) %>%
    pull(fraction)

  group2 <- df %>%
    filter(cell_type == cell & time == time2) %>%
    pull(fraction)

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

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

res_t_test
```

```
## # A tibble: 12 x 5
##    cell_type                    time_1 time_2 p_val              p_adj
##    <chr>                        <chr>  <chr>  <chr>              <dbl>
##  1 T cell CD4+ (non-regulatory) day7   day0   0.103492933593285  0.310
##  2 T cell regulatory (Tregs)    day7   day0   0.279300080913131  0.530
##  3 T cell CD8+                  day7   day0   0.401313899684524  0.535
##  4 T cell CD4+ (non-regulatory) day14  day0   0.0182048942412375 0.212
##  5 T cell regulatory (Tregs)    day14  day0   0.610321922325501  0.667
##  6 T cell CD8+                  day14  day0   0.0353146174302014 0.212
##  7 T cell CD4+ (non-regulatory) day21  day0   0.353059297908559  0.530
##  8 T cell regulatory (Tregs)    day21  day0   0.960290547277272  0.960
##  9 T cell CD8+                  day21  day0   0.34625148468339   0.530
## 10 T cell CD4+ (non-regulatory) day28  day0   0.0582454485005906 0.233
## 11 T cell regulatory (Tregs)    day28  day0   0.611099982346776  0.667
## 12 T cell CD8+                  day28  day0   0.146916847732862  0.353
```

Including non-T cell types

```
df2 <- df %>%
  mutate(cell_type = factor(cell_type,
                            levels = names(colPals$cell_types))) %>%
  group_by(time, cell_type) %>%
  summarize_each(dplyr::funs(mean, sd, se=sd(.)/sqrt(n())), fraction)

ggplot(df2, aes(x=time, y=mean, group=cell_type, fill=cell_type)) +
  geom_bar(stat="identity", position = position_dodge(0.9), width = 0.7, size=1, color='black') +
  geom_errorbar(aes(ymin=mean, ymax=mean+se*1.96), width=.6, lwd=1, position = position_dodge(width = 0.9)) +
  scale_fill_manual(values = colPals$cell_types) +
  scale_y_continuous(limits = c(0,0.6),
                     breaks = seq(0,0.6,0.2),
                     expand = expansion(mult = c(.01, .05))) +
  xlab("") +
  ylab("Cell type fraction (RNAseq)") +
  theme_bw(base_size = 20) +
  theme(
    title = element_text(size=15),
    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 = 'top',
    legend.title = element_blank(),
    legend.justification = 'top',
    legend.text = element_text(size=10)
  )
```



```
ggsave(filename = "plots/figS5_cell_type_deconvolution_all.pdf", width = 8, height = 5, units = "in", dpi = 300, device = cairo_pdf)
```

Exports

```
openxlsx::write.xlsx(
  list(cell_type_fractions = df,
       t_tests = res_t_test),
  file = "tables/dataS2_RNAseq_deconvolution.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] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] broom_1.0.4        RColorBrewer_1.1-3 immunedeconv_2.1.0 EPIC_1.1.6
##  [5] patchwork_1.1.2    magrittr_2.0.3     forcats_1.0.0      stringr_1.5.0
##  [9] dplyr_1.1.1        purrr_1.0.1        readr_2.1.4        tidyr_1.3.0
## [13] tibble_3.2.1       ggplot2_3.4.2      tidyverse_1.3.2
```

```
##
## loaded via a namespace (and not attached):
##   [1] googledrive_2.1.0           colorspace_2.1-0
##   [3] XVector_0.38.0              GenomicRanges_1.50.2
##   [5] fs_1.6.1                    rstudioapi_0.14
##   [7] farver_2.1.1                ComICS_1.0.4
##   [9] bit64_4.0.5                 AnnotationDbi_1.60.2
##  [11] fansi_1.0.4                 lubridate_1.9.2
##  [13] xml2_1.3.3                  codetools_0.2-18
##  [15] splines_4.2.1               cachem_1.0.7
##  [17] knitr_1.42                  jsonlite_1.8.4
##  [19] annotate_1.76.0             dbplyr_2.3.2
##  [21] png_0.1-8                   data.tree_1.0.0
##  [23] compiler_4.2.1              httr_1.4.5
##  [25] backports_1.4.1             Matrix_1.5-3
##  [27] fastmap_1.1.1               gargle_1.3.0
##  [29] limma_3.54.2                cli_3.6.1
##  [31] htmltools_0.5.5             prettyunits_1.1.1
##  [33] tools_4.2.1                 gtable_0.3.3
##  [35] glue_1.6.2                  GenomeInfoDbData_1.2.9
##  [37] quantiseqr_1.6.0            rappdirs_0.3.3
##  [39] Rcpp_1.0.10                 limSolve_1.5.6
##  [41] Biobase_2.58.0              cellranger_1.1.0
##  [43] vctrs_0.6.1                 Biostrings_2.66.0
##  [45] preprocessCore_1.60.2       nlme_3.1-157
##  [47] xfun_0.38                   openxlsx_4.2.5.1
##  [49] rvest_1.0.3                 lpSolve_5.6.18
##  [51] timechange_0.2.0            lifecycle_1.0.3
##  [53] XML_3.99-0.14               googlesheets4_1.1.0
##  [55] edgeR_3.40.2                zlibbioc_1.44.0
##  [57] MASS_7.3-57                 scales_1.2.1
##  [59] MatrixGenerics_1.10.0       hms_1.1.3
##  [61] SummarizedExperiment_1.28.0 parallel_4.2.1
##  [63] yaml_2.3.7                  curl_5.0.0
##  [65] memoise_2.0.1               biomaRt_2.46.3
##  [67] stringi_1.7.12              RSQLite_2.3.0
##  [69] highr_0.10                  genefilter_1.80.3
##  [71] S4Vectors_0.36.2            BiocGenerics_0.44.0
##  [73] zip_2.2.2                   BiocParallel_1.32.6
##  [75] testit_0.13                 GenomeInfoDb_1.34.9
##  [77] rlang_1.1.0                 pkgconfig_2.0.3
##  [79] bitops_1.0-7                matrixStats_0.63.0
##  [81] evaluate_0.20               lattice_0.20-45
##  [83] bit_4.0.5                   tidyselect_1.2.0
##  [85] R6_2.5.1                    IRanges_2.32.0
##  [87] generics_0.1.3              DelayedArray_0.24.0
##  [89] DBI_1.1.3                   pillar_1.9.0
##  [91] haven_2.5.2                 withr_2.5.0
##  [93] mgcv_1.8-40                 survival_3.3-1
##  [95] KEGGREST_1.38.0             RCurl_1.98-1.12
##  [97] modelr_0.1.11               crayon_1.5.2
##  [99] utf8_1.2.3                  BiocFileCache_1.14.0
## [101] tzdb_0.3.0                  rmarkdown_2.21
## [103] mMCPcounter_1.1.0           progress_1.2.2
## [105] locfit_1.5-9.7              grid_4.2.1
## [107] readxl_1.4.2                sva_3.46.0
## [109] blob_1.2.4                  reprex_2.0.2
## [111] digest_0.6.31               xtable_1.8-4
## [113] openssl_2.0.6               stats4_4.2.1
## [115] munsell_0.5.0               quadprog_1.5-8
## [117] askpass_1.1
```