Step 0: Pre-processing with all volunteers

Carlos Gallardo & Christian Oertlin

17 April, 2023

```
# Import libraries and helper functions
source("code/helper_functions.R")
library(tidyverse)
library(magrittr)
library(patchwork)
library(DESeq2)
library(limma)
library(variancePartition)
library(RColorBrewer)
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$factors <- setNames(c('#E31D27','#FA9F1C','#9A509F','#C1C1C1'),
                            nm = c("volunteer","time","batch","Residuals"))
```

Important note:

This script contains the exploratory analysis conducted on all 10 volunteers participating in the dry immersion (DI) study. As shown below, v9 and v10 specifically show elevated expression of monocyte genes (e.g., CDKN1A, NR4A1, NR4A2, MYADM, IRS2 and CD83) suggesting contamination for these samples. As per our exclusion criteria we chose to remove these for further analyses. The data provided includes volunteers 1 to 8 which are used for investigation in the study. Data with v9 and v10 can be provided upon request to reproduce the pre-processing steps in this script.

Load data

Background annotation

```
ann_data <- read.table(
  file = 'data/resources/gene_annotation_ensembl_v104.txt',
  stringsAsFactors = FALSE,
  sep = "\t",
  header = TRUE,
  fill = FALSE,
  quote = "") %>%
  dplyr::rename(Geneid = ensembl_gene_id)
```

RNAseq expression data

```
RNAseq <- vector("list")

RNAseq[["unfilt"]][["rawdata"]] <- read.table(file = 'data/rnaseq/exon_counts_all.txt',
                                              stringsAsFactors = FALSE,
```

```
                                          sep = "\t",
                                          header = TRUE)

# Make duplicated gene names unique
RNAseq[["unfilt"]][["rawdata"]] <- RNAseq[["unfilt"]][["rawdata"]] %>%
  mutate(GeneSymbol = uniquify(plyr::mapvalues(.$Geneid,
                                               from = ann_data$Geneid,
                                               to = ann_data$external_gene_name,
                                               warn_missing = F), sep = '_'))

RNAseq[["unfilt"]][["annotation"]] <- RNAseq[["unfilt"]][["rawdata"]] %>%
  select(Geneid, GeneSymbol) %>%
  inner_join(ann_data, by = "Geneid") %>%
  select(-external_gene_name)

RNAseq[["unfilt"]][["design"]] <- read.table(file = 'data/RNAseq/design_mtx_all.txt',
                                             stringsAsFactors = FALSE,
                                             sep = "\t",
                                             header = TRUE) %>%
  mutate(sample = factor(sample, levels = sample),
         batch = factor(batch, levels = unique(batch)),
         volunteer = factor(volunteer, levels = unique(volunteer)),
         time = factor(time, levels = unique(time)))

RNAseq[["unfilt"]][["counts"]] <- RNAseq[["unfilt"]][["rawdata"]] %>%
  select(-c(1:3)) %>%
  column_to_rownames("GeneSymbol")
```

Pre-processing

Filtering zero count genes

```
paste("Raw feature count:", nrow(RNAseq$unfilt$counts))
```

```
## [1] "Raw feature count: 60649"
```

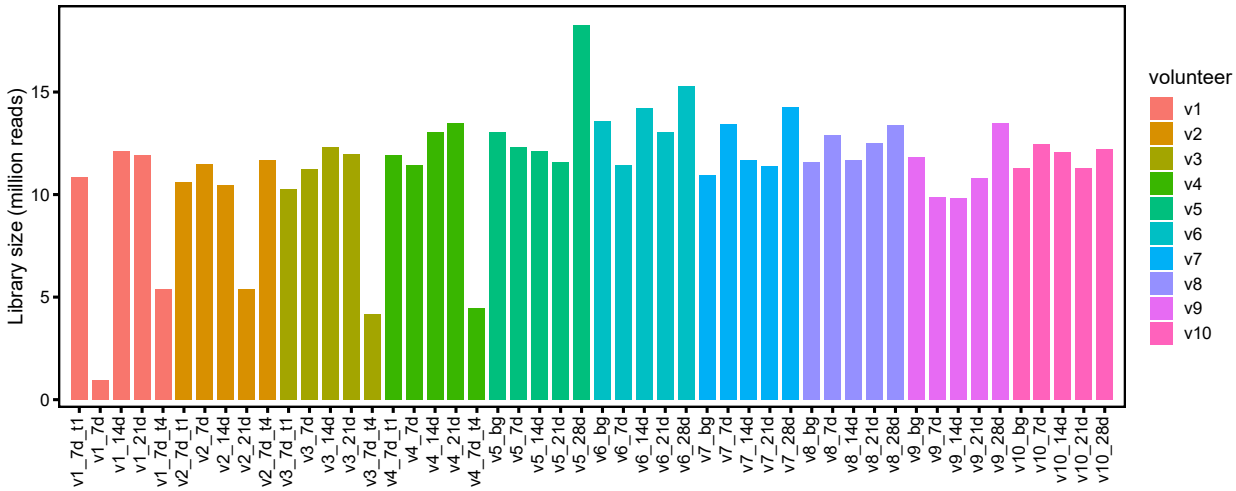
```
tokeep <- rowSums(RNAseq$unfilt$counts) > 0
paste("Non-zero feature count:", sum(tokeep))
```

```
## [1] "Non-zero feature count: 42934"
```

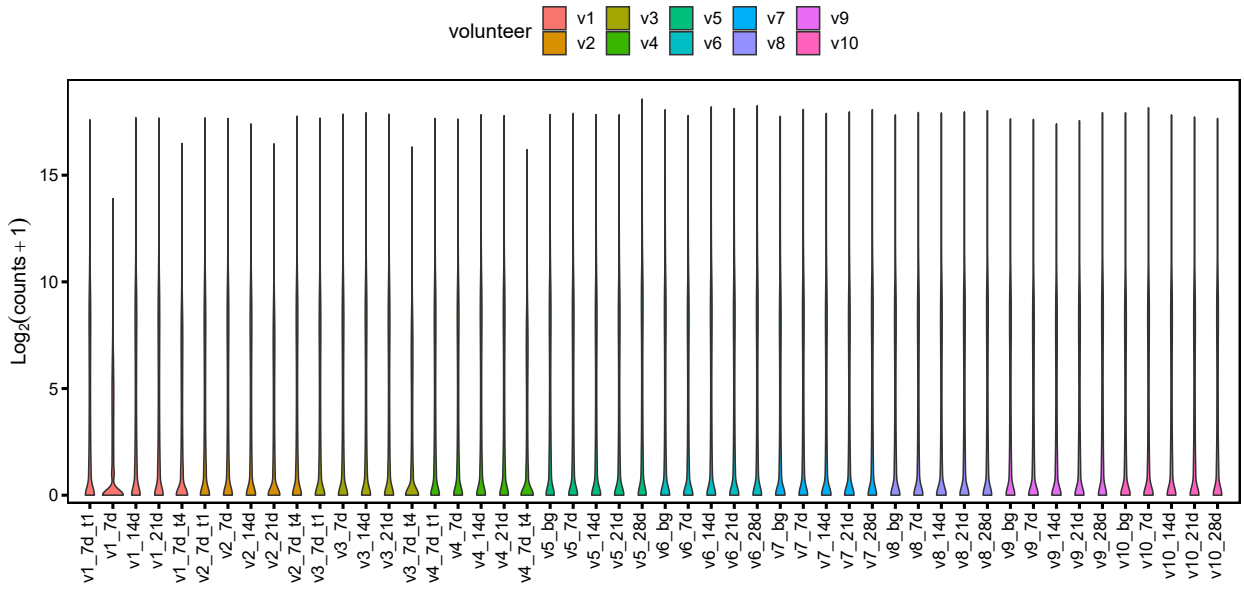
```
RNAseq$unfilt$rawdata <- RNAseq$unfilt$rawdata[tokeep,]
RNAseq$unfilt$annotation <- RNAseq$unfilt$annotation[tokeep,]
RNAseq$unfilt$counts <- RNAseq$unfilt$counts[tokeep,]
rm(tokeep)
```

Library sizes and gene expression distributions

```
df <- RNAseq$unfilt$design
df$lib.size <- colSums(RNAseq$unfilt$counts)

ggplot(df, aes(x=sample, y=lib.size/1e6, fill=volunteer)) +
  geom_bar(stat = "identity", width = 0.8) +
  xlab("") +
  ylab("Library size (million reads)") +
  scale_x_discrete(expand =expansion(mult = c(.02, .02))) +
  scale_y_continuous(expand =expansion(mult = c(.02, .05))) +
  theme_custom(
    axis.text.x.bottom = element_text(angle = 90, hjust = 1, vjust = 0.3)
  )
```

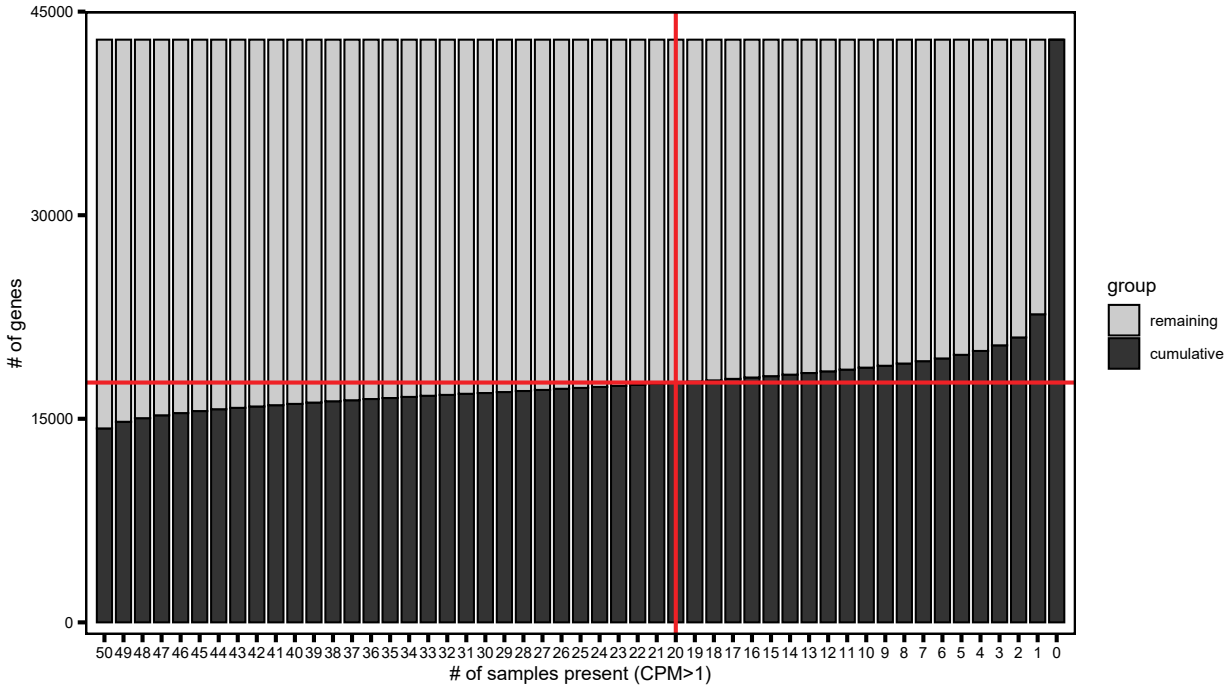



```
df <- RNAseq$unfilt$counts %>%
  rownames_to_column(var = "geneID") %>%
  pivot_longer(cols = c(2:length(.)),
               names_to = "sample") %>%
  dplyr::rename(counts = value)

df$volunteer <- rep(RNAseq$unfilt$design$volunteer, dim(RNAseq$unfilt$counts)[1])
df$time <- rep(RNAseq$unfilt$design$time, dim(RNAseq$unfilt$counts)[1])
df$sample <- factor(df$sample, levels = names(RNAseq$unfilt$counts))

ggplot(df, aes(x=sample, y=log2(counts+1), fill=volunteer)) +
  geom_violin(scale = "area") +
  xlab("") +
  ylab(expression(Log[2](counts+1))) +
  scale_x_discrete(expand =expansion(mult = c(.02, .02))) +
  scale_y_continuous(expand =expansion(mult = c(.02, .05))) +
  theme_custom(
    axis.text.x.bottom = element_text(angle = 90, hjust = 1, vjust = 0.3),
    legend.position = "top"
  )
```



Filtering low abundance genes

```
# Calculate CPM
RNAseq[["unfilt"]][["cpm"]] <- cpm.normalize(RNAseq$unfilt$counts)

abovethresh <- RNAseq$unfilt$cpm > 1

df <- data.frame(samples = factor(seq(0,ncol(RNAseq$unfilt$cpm),1),
                                  levels = rev(seq(0,ncol(RNAseq$unfilt$cpm),1))),
                 genes = c(table(rowSums(abovethresh)))) %>%
  mutate(cumulative = rev(cumsum(rev(genes)))) %>%
  mutate(remaining = sum(genes)-cumulative) %>%
  pivot_longer(cols = c("cumulative","remaining"),
               names_to = "group") %>%
  mutate(group = factor(group, levels = c("remaining", "cumulative")))

ggplot(data=df, aes(x=samples, y=value, fill=group)) +
  geom_bar(color="black", size=0.5, width=0.8, position="stack", stat="identity") +
  geom_hline(yintercept = unlist(df[df$samples == "20" & df$group == "cumulative", "value"]),
             linetype="solid", size=1, color="#EF2126") +
  geom_vline(xintercept = "20", linetype="solid", size=1, color="#EF2126") +
  xlab("# of samples present (CPM>1)") +
  ylab("# of genes") +
  scale_x_discrete(expand =expansion(mult = c(.02, .02))) +
  scale_y_continuous(expand =expansion(mult = c(.02, .00)),
                     limits = c(0,45000), breaks = seq(0,45000,15000)) +
  scale_fill_manual(values = c("grey80", "grey20")) +
  theme_custom(base_size = 8)
```




```
tokeep <- rowSums(abovethresh) >= 20
paste("Pre-filtering gene count:", length(tokeep))
```

```
## [1] "Pre-filtering gene count: 42934"
```

```
paste("Genes below abundance threshold:", length(tokeep)-sum(tokeep))
```

```
## [1] "Genes below abundance threshold: 25264"
```

```
paste("Remaining genes:", sum(tokeep))
```

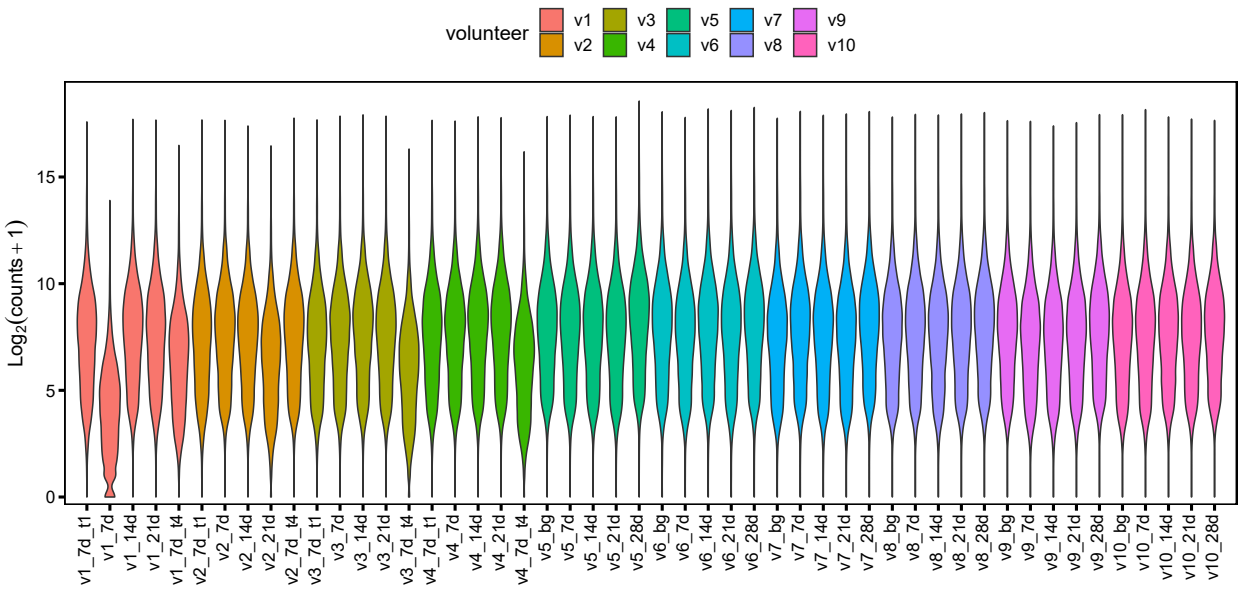
```
## [1] "Remaining genes: 17670"
```

```
# Filter genes
RNAseq[["filt"]][["rawdata"]] <- RNAseq$unfilt$rawdata[tokeep,]
RNAseq[["filt"]][["annotation"]] <- RNAseq$unfilt$annotation[tokeep,]
RNAseq[["filt"]][["design"]] <- RNAseq$unfilt$design
RNAseq[["filt"]][["counts"]] <- RNAseq$unfilt$counts[tokeep,]
```

```
# Normalize
RNAseq[["filt"]][["cpm"]] <- cpm.normalize(RNAseq$filt$counts)
```

Gene expression distribution post-filtering

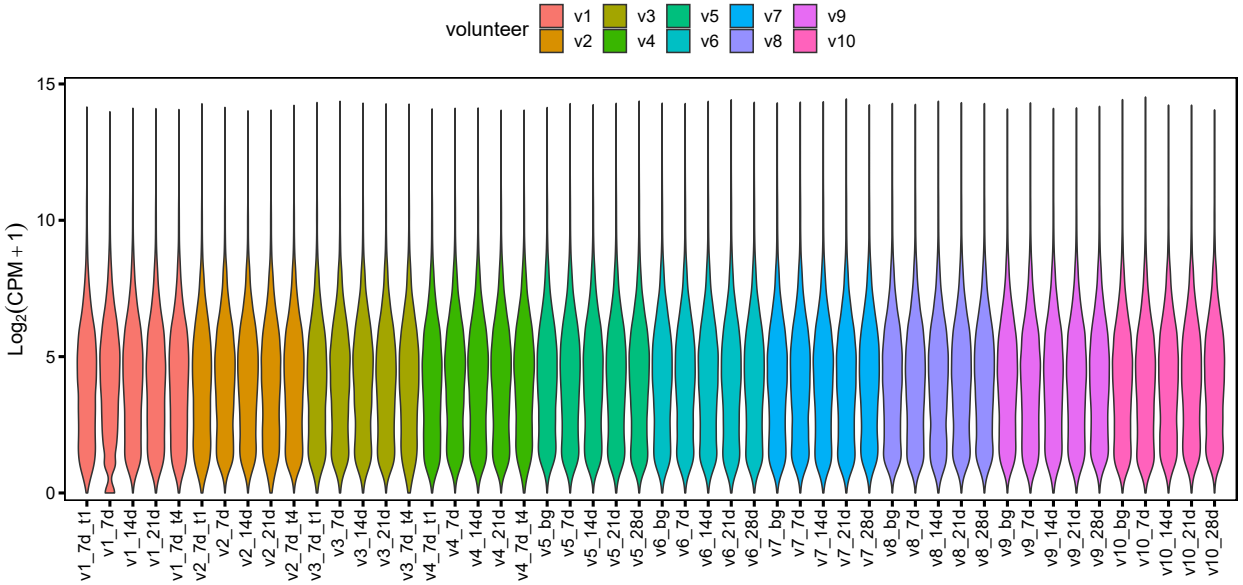
```
# Expression distribution post-filtering
df <- RNAseq$filt$counts %>%
  rownames_to_column(var = "geneID") %>%
  pivot_longer(cols = c(2:length(.)),
               names_to = "sample") %>%
  dplyr::rename(counts = value)

df$volunteer <- rep(RNAseq$filt$design$volunteer, dim(RNAseq$filt$counts)[1])
df$time <- rep(RNAseq$filt$design$time, dim(RNAseq$filt$counts)[1])
df$sample <- factor(df$sample, levels = names(RNAseq$filt$counts))

ggplot(df, aes(x=sample, y=log2(counts+1), fill=volunteer)) +
  geom_violin(scale = "area") +
  xlab("") +
  ylab(expression(Log[2](counts+1))) +
  scale_x_discrete(expand =expansion(mult = c(.02, .02))) +
  scale_y_continuous(expand =expansion(mult = c(.02, .05))) +
  theme_custom(
    axis.text.x.bottom = element_text(angle = 90, hjust = 1, vjust = 0.3),
    legend.position = "top"
  )
```



Normalized gene expression distribution post-filtering

```
# Normalized expression distribution post-filtering
df <- RNAseq$filt$counts %>%
  rownames_to_column(var = "geneID") %>%
  pivot_longer(cols = c(2:length(.)),
               names_to = "sample") %>%
  dplyr::rename(counts = value)

df$volunteer <- rep(RNAseq$filt$design$volunteer, dim(RNAseq$filt$counts)[1])
df$time <- rep(RNAseq$filt$design$time, dim(RNAseq$filt$counts)[1])
df$sample <- factor(df$sample, levels = names(RNAseq$filt$counts))

df$cpm <- RNAseq$filt$cpm %>%
  pivot_longer(cols = c(1:length(.)),
               names_to = "sample") %>%
  select(value) %>%
  unlist()
```

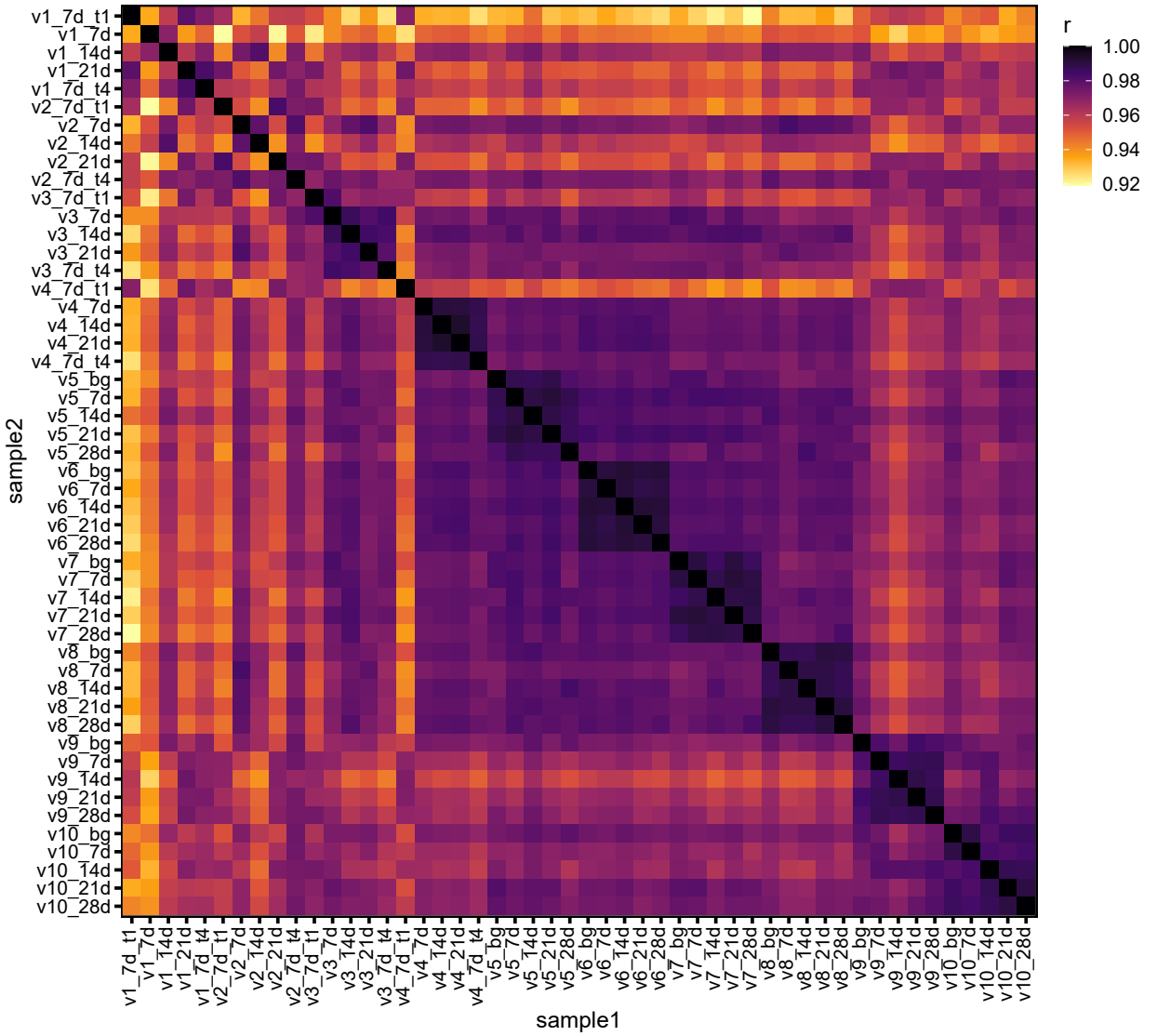
```
ggplot(df, aes(x=sample, y=log2(cpm+1), fill=volunteer)) +
  geom_violin(scale = "area") +
  xlab("") +
  ylab(expression(Log[2](CPM+1))) +
  scale_x_discrete(expand =expansion(mult = c(.02, .02))) +
  scale_y_continuous(expand =expansion(mult = c(.02, .05))) +
  theme_custom(
    axis.text.x.bottom = element_text(angle = 90, hjust = 1, vjust = 0.3),
    legend.position = "top"
  )
```



Transcriptome differences

Sample correlations

```
df <- cor(log2(RNAseq$filt$cpm+1), method = "spearman") %>%
  as.data.frame() %>%
  rownames_to_column(var = "sample1") %>%
  mutate(across(everything(), as.character)) %>%
  pivot_longer(cols = c(2:length(.)),
               names_to = "sample2") %>%
  dplyr::rename(r = value) %>%
  mutate(sample1 = factor(sample1, levels = names(RNAseq$filt$counts)),
         sample2 = factor(sample2, levels = names(RNAseq$filt$counts)),
         r = as.numeric(r))

ggplot(df, aes(x=sample1, y=sample2, fill= r)) +
  geom_tile() +
  scale_y_discrete(limits=rev) +
  scale_fill_gradientn(colours = rev(colPals$inferno)) +
  theme_custom(
    axis.text.x.bottom = element_text(angle = 90, hjust = 1, vjust = 0.3),
    legend.position = "right",
    legend.justification = "top"
  )
```



DESeq2 analysis

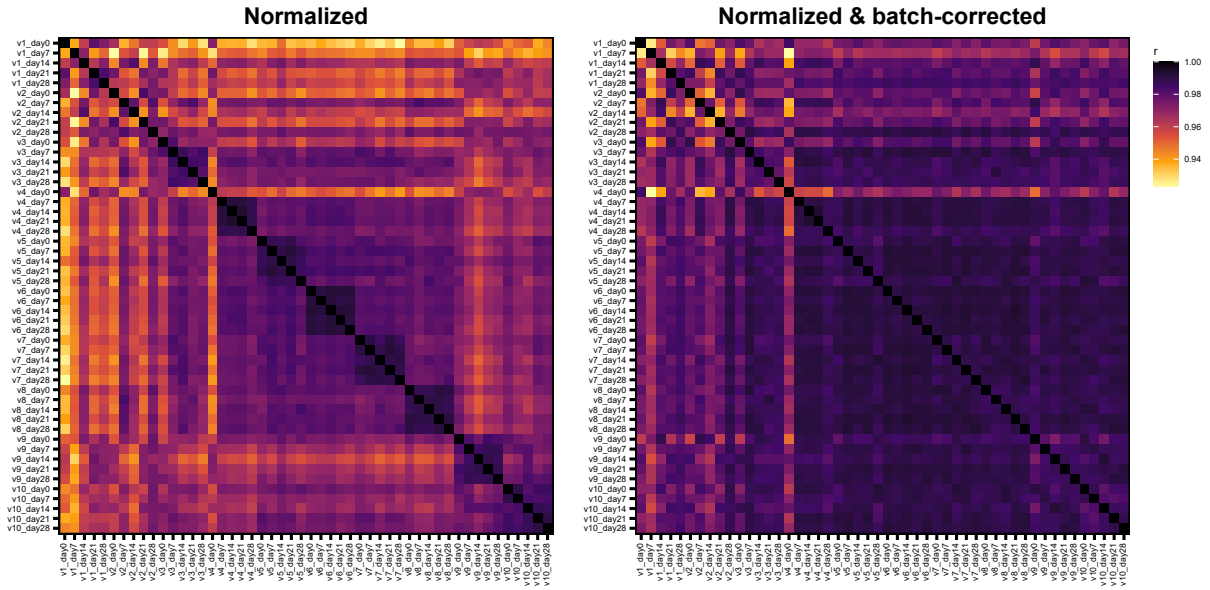
```
# DESeq2 pipeline
dsData <- DESeqDataSetFromMatrix(countData = RNAseq$filt$counts,
                                 colData = RNAseq$filt$design,
                                 design = ~volunteer + time)
dsData <- estimateSizeFactors(dsData)
dsData <- DESeq(dsData, test = "LRT", reduced = ~volunteer)

RNAseq$filt[["DESeq_norm"]] <- counts(dsData, normalized=TRUE) %>% as.data.frame()
RNAseq$filt[["DESeq_vst"]] <- assay(vst(dsData, blind=FALSE)) %>% as.data.frame()
RNAseq$filt[["DESeq_rlog"]] <- assay(rlog(dsData, blind=FALSE)) %>% as.data.frame()

DESeq2_DEGs <- list(
  GLMtime = results(dsData, test = "LRT", independentFiltering = T),
  day7Vsday0 = results(dsData, contrast=c("time", "day7", "day0"), test = "Wald", independentFiltering = T),
  day14Vsday0 = results(dsData, contrast=c("time", "day14", "day0"), test = "Wald", independentFiltering = T),
  day21Vsday0 = results(dsData, contrast=c("time", "day21", "day0"), test = "Wald", independentFiltering = T),
  day28Vsday0 = results(dsData, contrast=c("time", "day28", "day0"), test = "Wald", independentFiltering = T)
)
DESeq2_DEGs <- lapply(DESeq2_DEGs, as.data.frame)
DESeq2_DEGs$GLMtime <- DESeq2_DEGs$GLMtime %>%
  add_column(logFC_day7Vsday0 = DESeq2_DEGs$day7Vsday0$log2FoldChange, .before = "log2FoldChange") %>%
  add_column(logFC_day14Vsday0 = DESeq2_DEGs$day14Vsday0$log2FoldChange, .before = "log2FoldChange") %>%
  add_column(logFC_day21Vsday0 = DESeq2_DEGs$day21Vsday0$log2FoldChange, .before = "log2FoldChange") %>%
```

```
  add_column(logFC_day28Vsday0 = DESeq2_DEGs$day28Vsday0$log2FoldChange, .before = "log2FoldChange") %>%
  select(-log2FoldChange)
DESeq2_DEGs <- lapply(DESeq2_DEGs, function(x) mutate(x, padj=ifelse(is.na(padj), 1, padj)))
DESeq2_DEGs <- lapply(DESeq2_DEGs, function(x) arrange(x, padj))
DESeq2_DEGs <- lapply(DESeq2_DEGs, rownames_to_column, var = "GeneSymbol")
DESeq2_DEGs <- lapply(DESeq2_DEGs, inner_join, y = RNAseq$filt$annotation, by = "GeneSymbol")

DESeq2_DEGs_filt <- list(
  padj_02 = lapply(DESeq2_DEGs, function(x) x %>% filter(padj<0.2)),
  padj_005 = lapply(DESeq2_DEGs, function(x) x %>% filter(padj<0.05)),
  padj_001 = lapply(DESeq2_DEGs, function(x) x %>% filter(padj<0.01))
)
```

Batch correction

```
# Remove batch effects from data
design_mtx <- model.matrix(~time, data = RNAseq$filt$design)
RNAseq[["filt"]][["DESeq_vst_nobatch"]] <- limma::removeBatchEffect(RNAseq$filt$DESeq_vst,
                                                                    batch = dsData$volunteer,
                                                                    design = design_mtx) %>% as.data.frame()
RNAseq[["filt"]][["DESeq_rlog_nobatch"]] <- limma::removeBatchEffect(RNAseq$filt$DESeq_rlog,
                                                                     batch = dsData$volunteer,
                                                                     design = design_mtx) %>% as.data.frame()

# Plot sample correlations
df <- cor(RNAseq$filt$DESeq_vst, method = "spearman") %>%
  as.data.frame() %>%
  rownames_to_column(var = "sample1") %>%
  mutate(across(everything(), as.character)) %>%
  pivot_longer(cols = c(2:length(.)),
               names_to = "sample2") %>%
  dplyr::rename(r = value) %>%
  mutate(sample1 = factor(sample1, levels = names(RNAseq$filt$counts)),
         sample2 = factor(sample2, levels = names(RNAseq$filt$counts)),
         r = as.numeric(r))

p1 <- ggplot(df, aes(x=sample1, y=sample2, fill= r)) +
  geom_tile() +
  scale_x_discrete(labels=paste(RNAseq$filt$design$volunteer,
                                RNAseq$filt$design$time,
                                sep = '_')) +
  scale_y_discrete(limits=rev, labels=rev(paste(RNAseq$filt$design$volunteer,
                                                RNAseq$filt$design$time,
                                                sep = '_'))) +
  scale_fill_gradientn(colours = rev(colPals$inferno)) +
  xlab('') +
  ylab('') +
  ggtitle('Normalized') +
  theme_custom(
    base_size = 6,
    axis.text.x.bottom = element_text(angle = 90, hjust = 1, vjust = 0.3),
    legend.position = "none",
    plot.title = element_text(size=16, face='bold', hjust=0.5)
  )

df2 <- cor(RNAseq$filt$DESeq_vst_nobatch, method = "spearman") %>%
  as.data.frame() %>%
  rownames_to_column(var = "sample1") %>%
  mutate(across(everything(), as.character)) %>%
  pivot_longer(cols = c(2:length(.)),
               names_to = "sample2") %>%
  dplyr::rename(r = value) %>%
  mutate(sample1 = factor(sample1, levels = names(RNAseq$filt$counts)),
         sample2 = factor(sample2, levels = names(RNAseq$filt$counts)),
         r = as.numeric(r))

p2 <- ggplot(df2, aes(x=sample1, y=sample2, fill= r)) +
  geom_tile() +
  scale_x_discrete(labels=paste(RNAseq$filt$design$volunteer,
                                RNAseq$filt$design$time,
                                sep = '_')) +
  scale_y_discrete(limits=rev, labels=rev(paste(RNAseq$filt$design$volunteer,
                                                RNAseq$filt$design$time,
                                                sep = '_'))) +
  scale_fill_gradientn(colours = rev(colPals$inferno)) +
  xlab('') +
  ylab('') +
  ggtitle('Normalized & batch-corrected') +
```

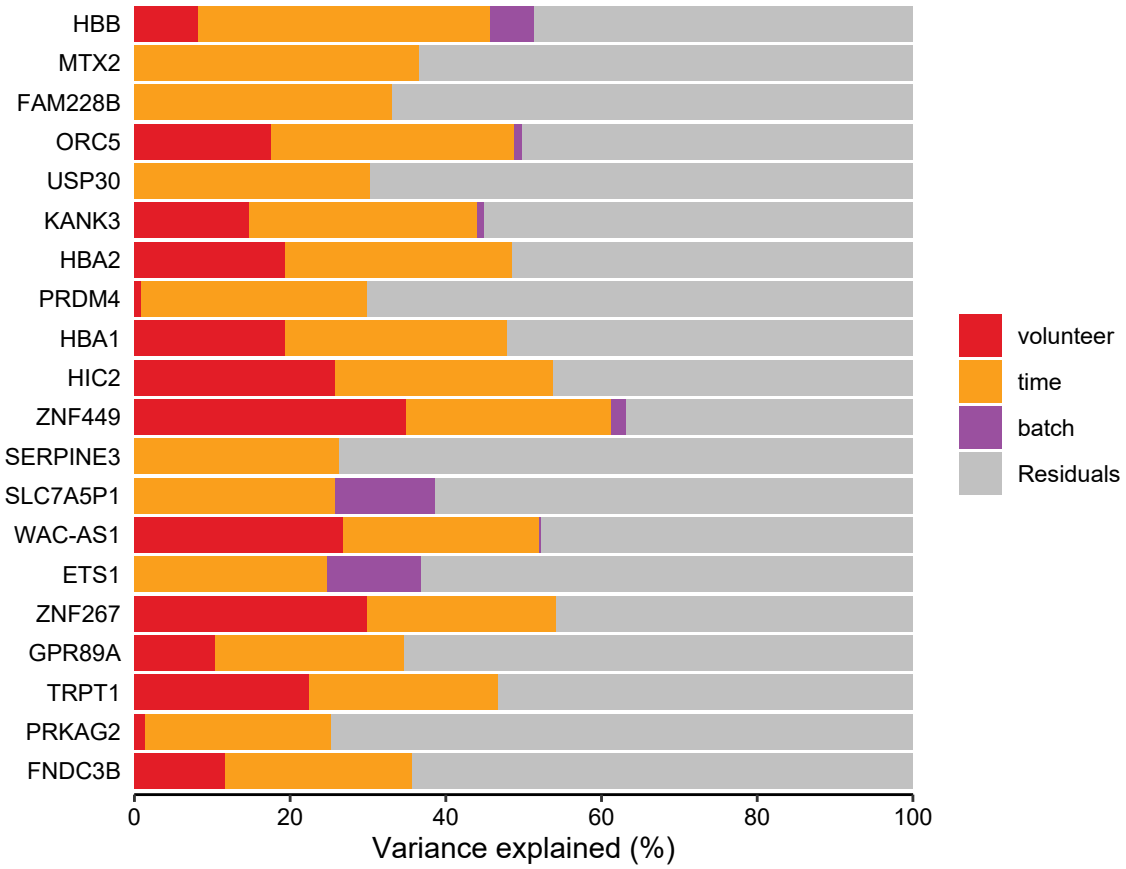
```
  theme_custom(
    base_size = 6,
    axis.text.x.bottom = element_text(angle = 90, hjust = 1, vjust = 0.3),
    legend.position = "right",
    legend.justification = "top",
    plot.title = element_text(size=16, face='bold', hjust=0.5)
  )

p1 + p2
```



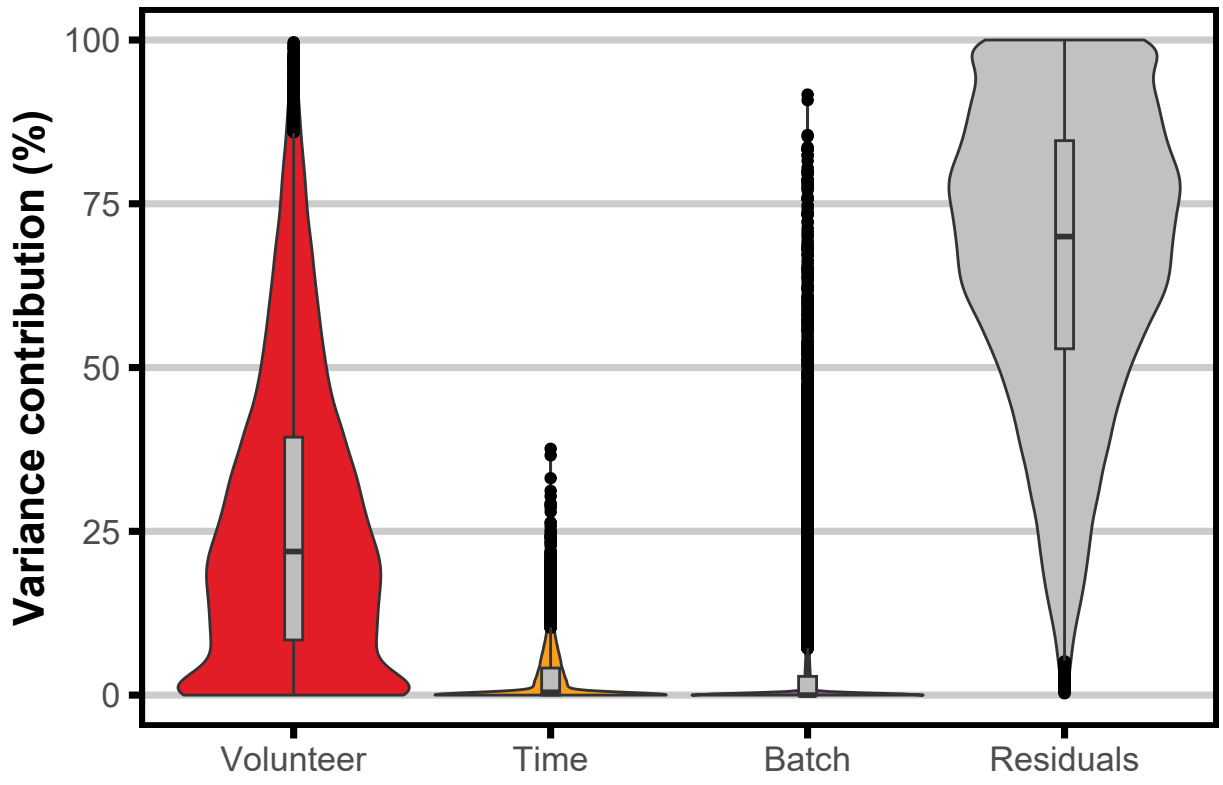
Variance partition

Without batch-correction

```
varPart <- fitExtractVarPartModel(as.matrix(RNAseq$filt$DESeq_vst),
                                  ~(1|batch) +(1|volunteer) + (1|time),
                                  RNAseq$filt$design)

vp <- sortCols( varPart ) %>%
  as.data.frame() %>%
  arrange(desc(`time`))

plotPercentBars(vp[1:20,]) +
  scale_fill_manual(values = colPals$factors)
```

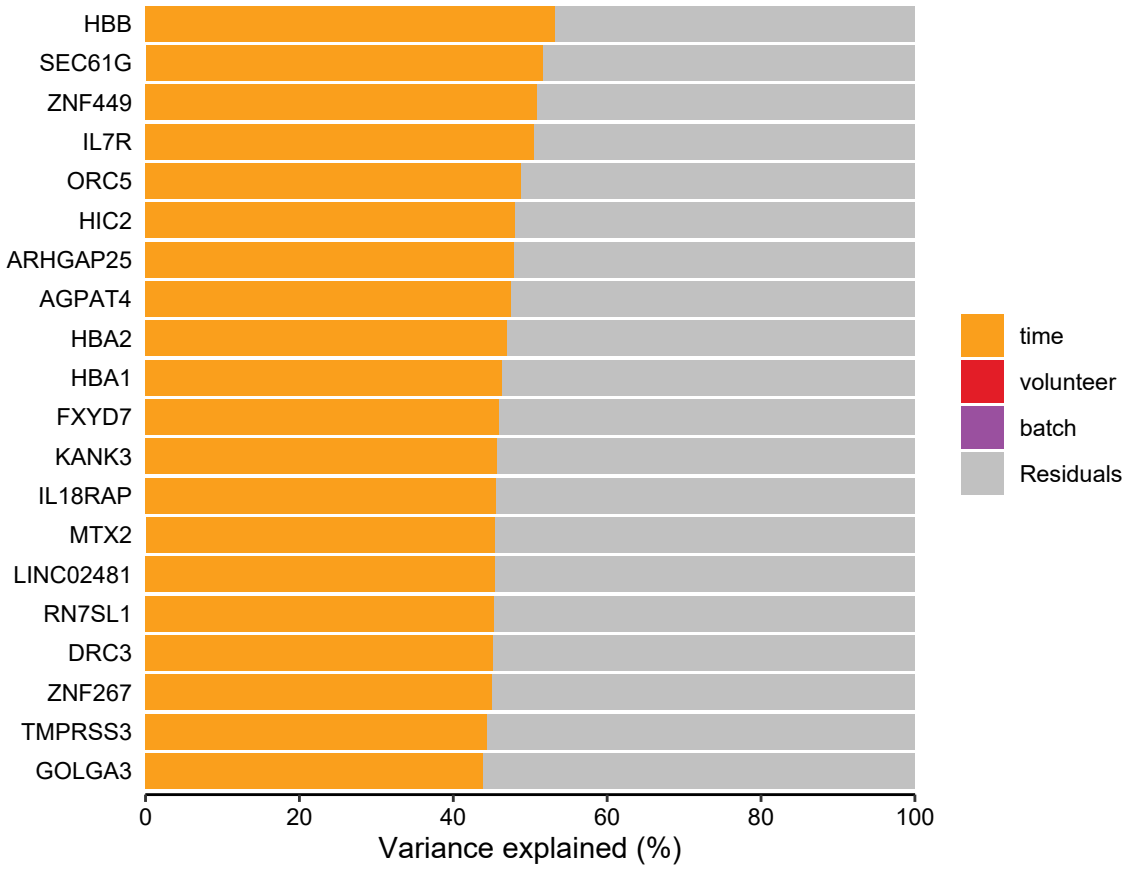



```
p3 <- plotVarPart(vp) +
  scale_fill_manual(values = colPals$factors) +
  scale_x_discrete(labels=c('Volunteer','Time','Batch','Residuals')) +
  ylab('Variance contribution (%)') +
  labs(fill = "Factor") +
  theme_bw(base_size = 16) +
  theme(
    legend.position = "none",
    axis.title.y = element_text(size=16, face='bold'),
    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)
  )

p3
```

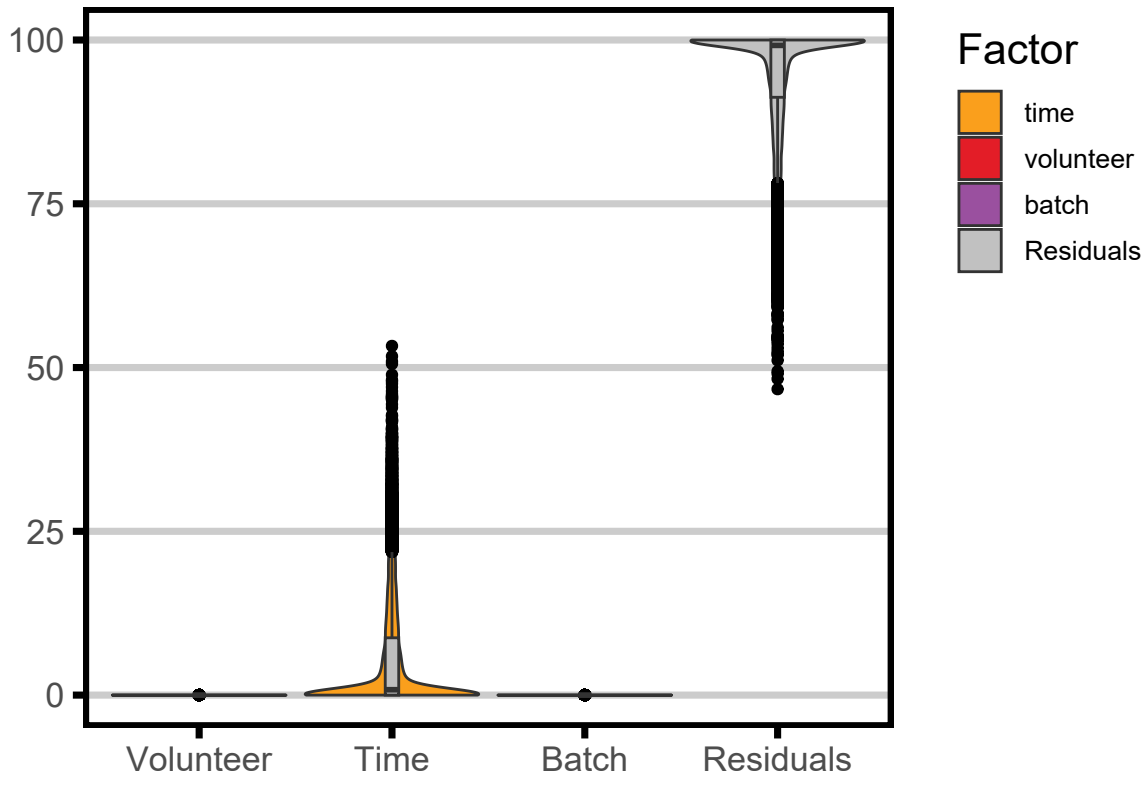


With batch-correction

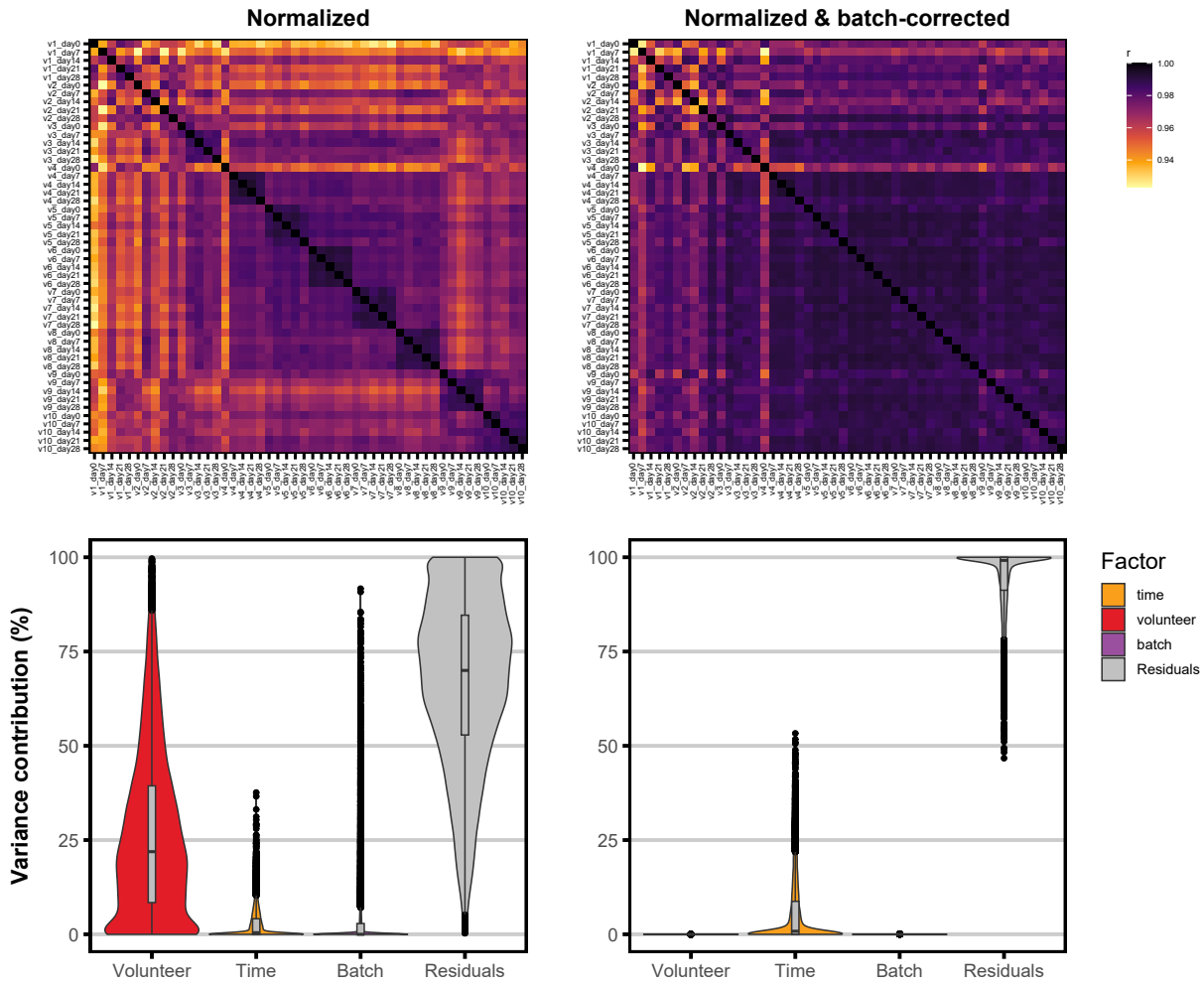
```
varPart <- fitExtractVarPartModel(as.matrix(RNAseq$filt$DESeq_vst_nobatch),
                                  ~(1|batch) +(1|volunteer) + (1|time),
                                  RNAseq$filt$design)

vp <- sortCols( varPart ) %>%
  as.data.frame() %>%
  arrange(desc(`time`))

plotPercentBars(vp[1:20,]) +
  scale_fill_manual(values = colPals$factors)
```




```
p4 <- plotVarPart(vp) +
  scale_x_discrete(limits = factor(names(colPals$factors),
                                   levels = names(colPals$factors)),
                   labels = c('Volunteer','Time','Batch','Residuals')) +
  scale_fill_manual(values = colPals$factors) +
  labs(fill = "Factor") +
  theme_bw(base_size = 16) +
  theme(
    legend.position = "right",
    legend.justification = "top",
    legend.text = element_text(size = 10),
    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)
  )

p4
```




```
p <- patchwork::wrap_plots(p1,p2,p3,p4,ncol=2)

p
```



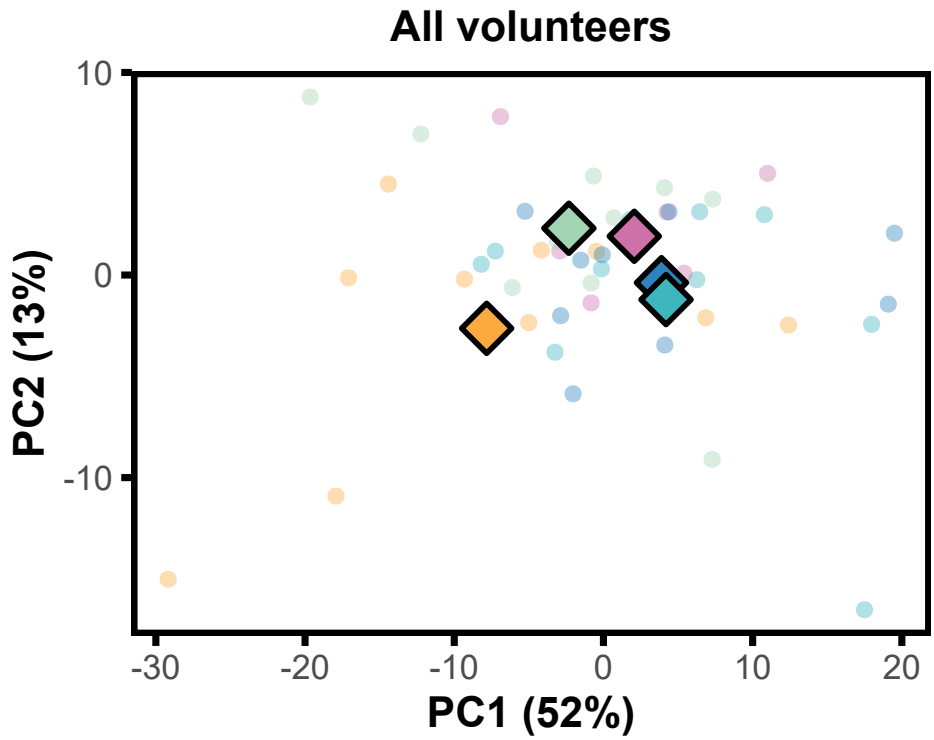

```
ggsave("plots/figS1_batch_correction.pdf", plot = p, width = 12, height = 10, units = "in", dpi = 300, device = cairo_pdf)
```

PCA plots

Mean

```
# PCA with mean summarisation
pca <- doPCA(RNAseq$filt$DESeq_vst_nobatch)

df <- pca$pcs %>%
  cbind(RNAseq$filt$design)

df2 <- df %>%
  group_by(time) %>%
  summarize(PC1 = mean(PC1),
            PC2 = mean(PC2),
            PC3 = mean(PC3),) %>%
  dplyr::rename(time2 = time)

ggplot() +
  geom_point(data = df, aes(x=PC1, y=PC2, color=time), shape=16, size=3, stroke=0, alpha=0.4) +
  geom_point(data = df2, aes(x=PC1, y=PC2, fill=time2), color="black", shape=23, size=6, stroke=1.5, alpha=1) +
  ggrepel::geom_label_repel() +
  xlab(paste("PC1 (", round(pca$percentVar[1],0), "%)", sep = "")) +
  ylab(paste("PC2 (", round(pca$percentVar[2],0), "%)", sep = "")) +
  scale_color_manual(values = colPals$time) +
  scale_fill_manual(values = colPals$time) +
  ggtitle('All volunteers') +
  theme_bw(base_size = 16) +
```

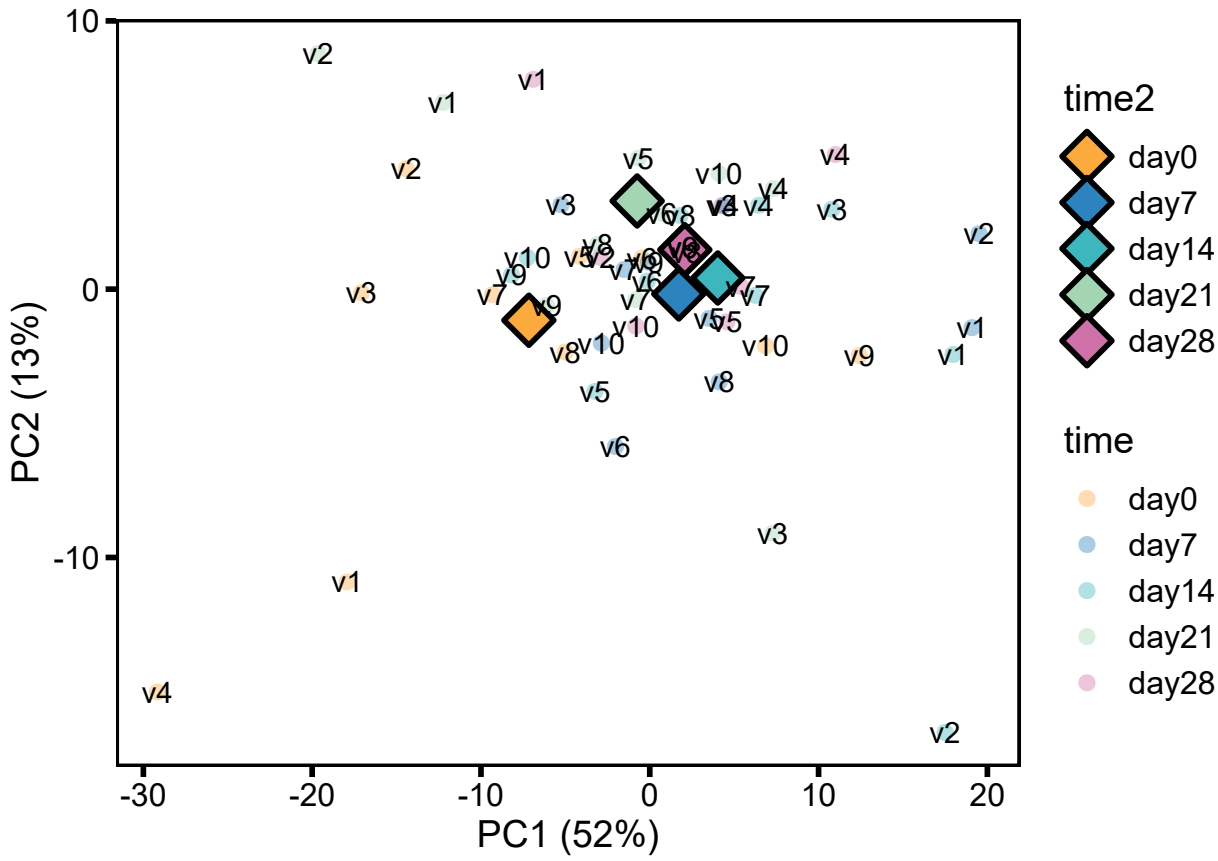
```
  theme(
    legend.position = "none",
    plot.title = element_text(size=16, face='bold', hjust=0.5),
    axis.title = element_text(size=16, face='bold'),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_rect(color = "black", fill = NA, size = 2),
    axis.ticks = element_line(color = "black", size = 1.25)
  )
```



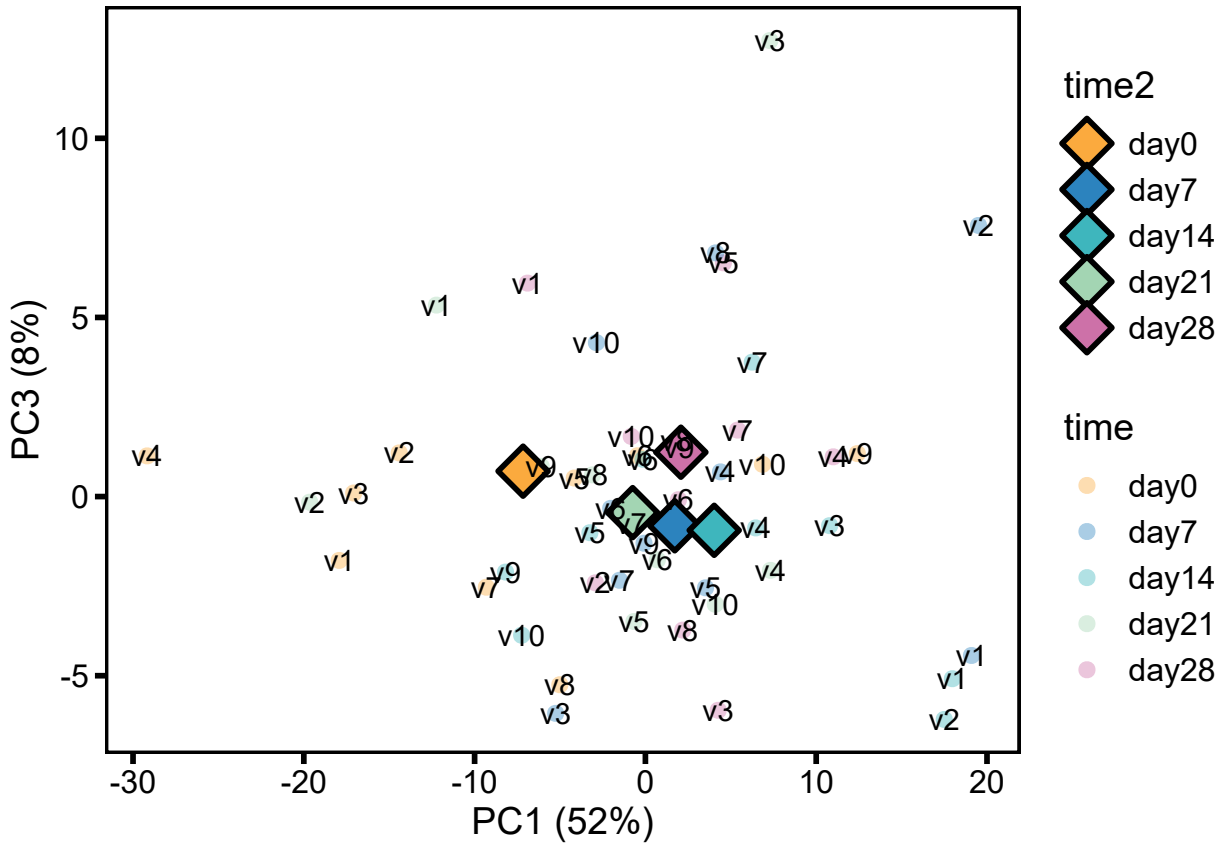

```
ggsave("plots/figS2_pca_all_volunteers.pdf", width = 5, height = 4, units = "in", dpi = 300, device = cairo_pdf)
```

Median

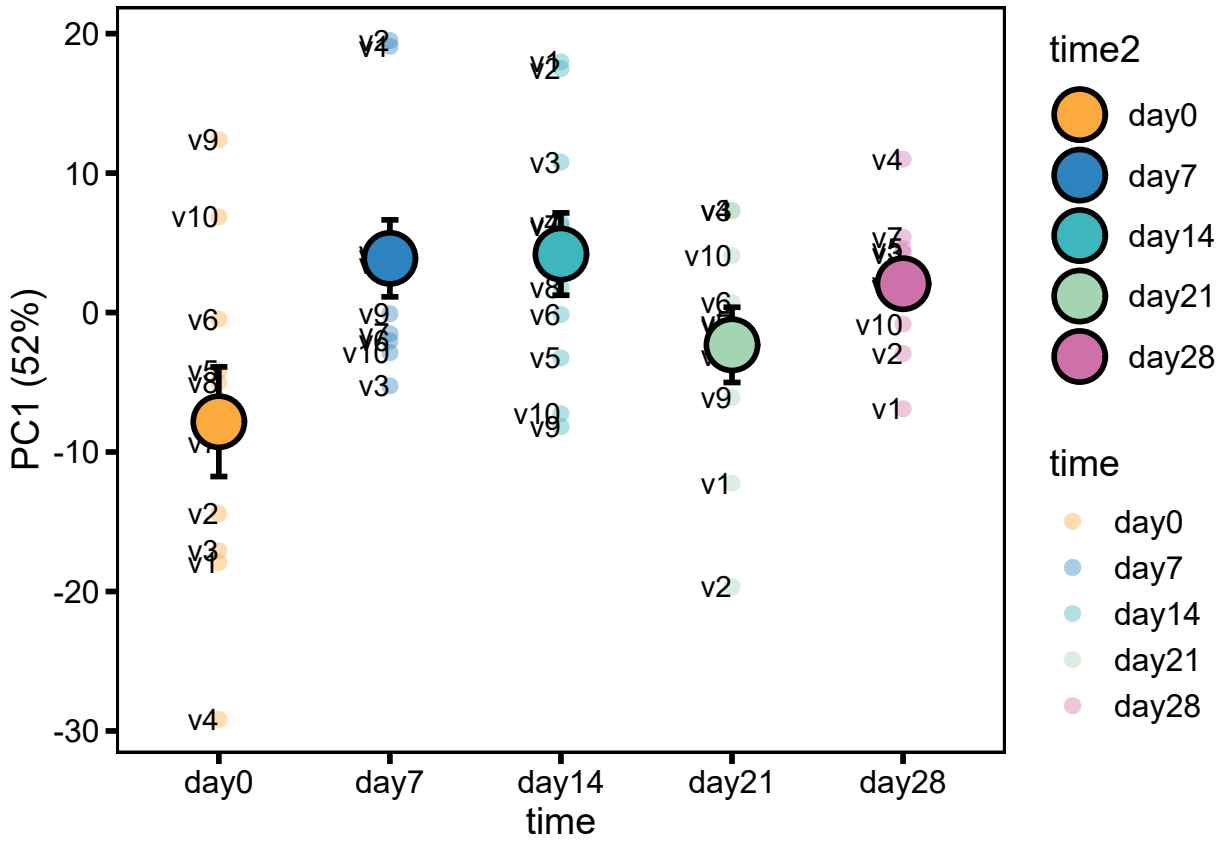
```
# PCA with median summarisation
pca <- doPCA(RNAseq$filt$DESeq_vst_nobatch)

df <- pca$pcs %>%
  cbind(RNAseq$filt$design)

df2 <- df %>%
  group_by(time) %>%
  summarize(PC1 = median(PC1),
            PC2 = median(PC2),
            PC3 = median(PC3),) %>%
  dplyr::rename(time2 = time)

ggplot() +
  geom_point(data = df, aes(x=PC1, y=PC2, color=time), shape=16, size=3, stroke=0, alpha=0.4) +
  geom_point(data = df2, aes(x=PC1, y=PC2, fill=time2), color="black", shape=23, size=6, stroke=1.5, alpha=1) +
  ggrepel::geom_label_repel() +
  theme_custom(legend.position = "right") +
  xlab(paste("PC1 (", round(pca$percentVar[1],0), "%)", sep = "")) +
  ylab(paste("PC2 (", round(pca$percentVar[2],0), "%)", sep = "")) +
  scale_color_manual(values = colPals$time) +
  scale_fill_manual(values = colPals$time) +
  geom_text(data = df,aes(x=PC1,y=PC2,label=volunteer))
```



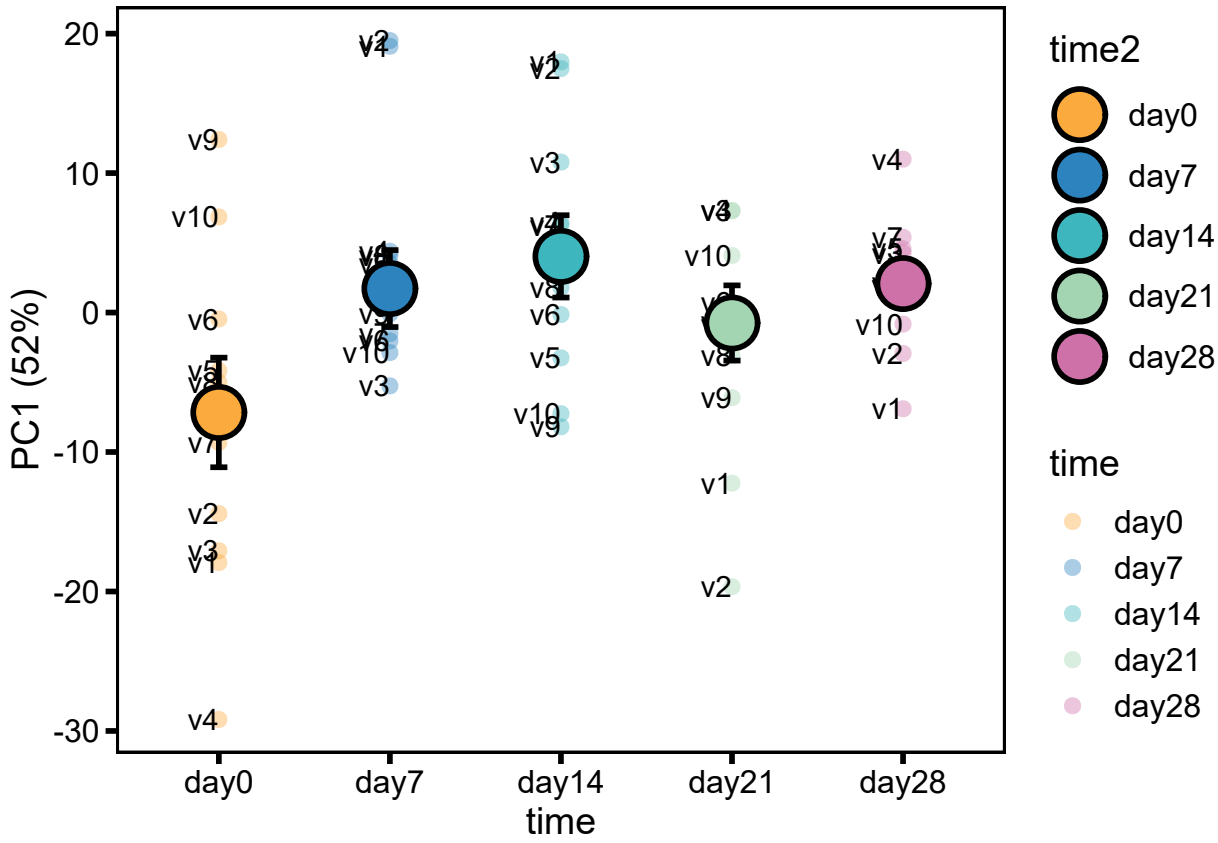

```
# PCA with median summarisation
pca <- doPCA(RNAseq$filt$DESeq_vst_nobatch)

df <- pca$pcs %>%
  cbind(RNAseq$filt$design)

df2 <- df %>%
  group_by(time) %>%
  summarize(PC1 = median(PC1),
            PC2 = median(PC2),
            PC3 = median(PC3),) %>%
  dplyr::rename(time2 = time)

ggplot() +
  geom_point(data = df, aes(x=PC1, y=PC3, color=time), shape=16, size=3, stroke=0, alpha=0.4) +
  geom_point(data = df2, aes(x=PC1, y=PC3, fill=time2), color="black", shape=23, size=6, stroke=1.5, alpha=1) +
  ggrepel::geom_label_repel() +
  theme_custom(legend.position = "right") +
  xlab(paste("PC1 (", round(pca$percentVar[1],0), "%)", sep = "")) +
  ylab(paste("PC3 (", round(pca$percentVar[3],0), "%)", sep = "")) +
  scale_color_manual(values = colPals$time) +
  scale_fill_manual(values = colPals$time) +
  geom_text(data = df,aes(x=PC1,y=PC3,label=volunteer))
```



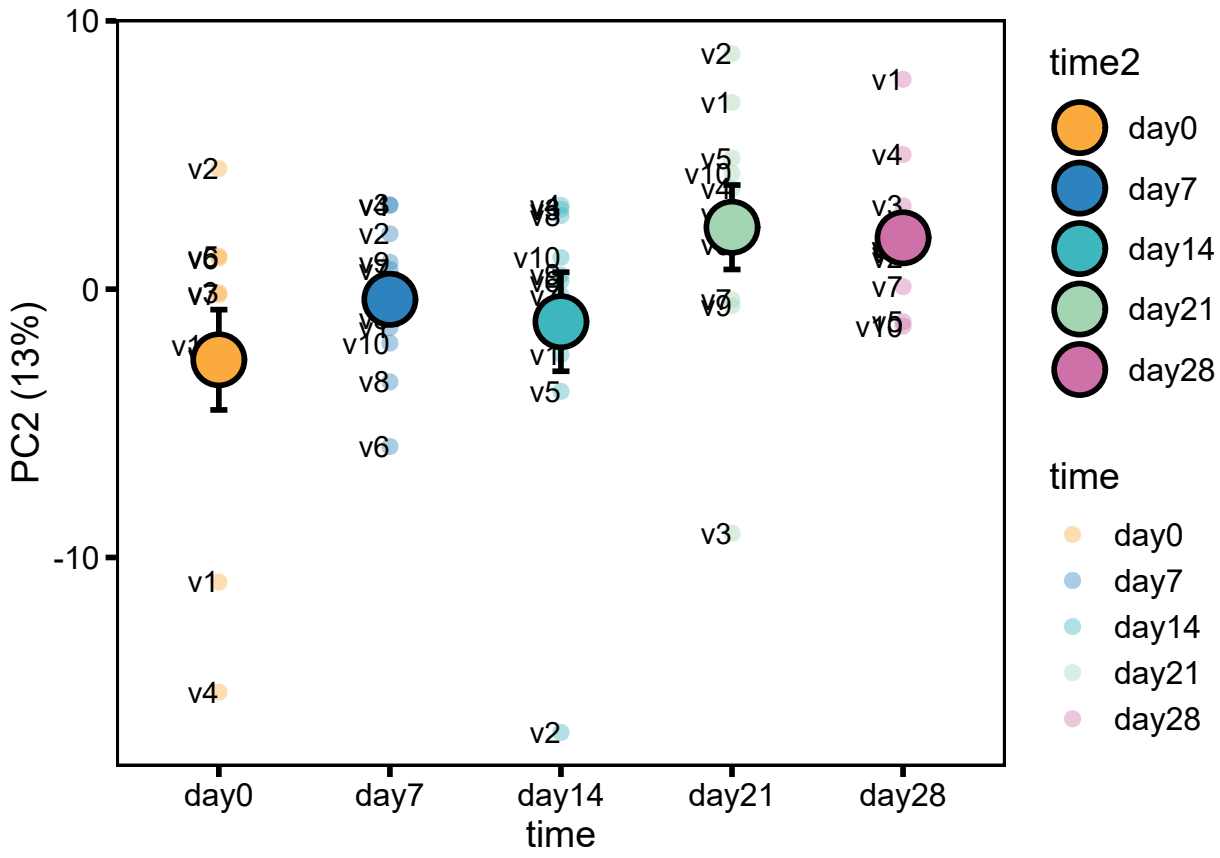

```
# boxplot with mean summarisation PC1
pca <- doPCA(RNAseq$filt$DESeq_vst_nobatch)

df <- pca$pcs %>%
  cbind(RNAseq$filt$design)

df2 <- df %>%
  group_by(time) %>%
  select(time, PC1) %>%
  summarize_each(dplyr::funs(mean, sd, se=sd(.)/sqrt(n())), PC1) %>%
  dplyr::rename(time2 = time)

ggplot() +
  geom_point(data = df, aes(x=time, y=PC1, color=time), shape=16, size=3, stroke=0, alpha=0.4) +
  geom_text(data = df,aes(x=time,y=PC1,label=volunteer),hjust=1)+
  geom_errorbar(data=df2, aes(x=time2, y=mean, ymin=mean-se, ymax=mean+se), width=.1, lwd=1) +
  geom_point(data = df2, aes(x=time2, y=mean, fill=time2), color="black", shape=21, size=8, stroke=1.5, alpha=1) +
  theme_custom(legend.position = "right") +
  ylab(paste("PC1 (", round(pca$percentVar[1],0), "%)", sep = "")) +
  scale_color_manual(values = colPals$time) +
  scale_fill_manual(values = colPals$time)
```



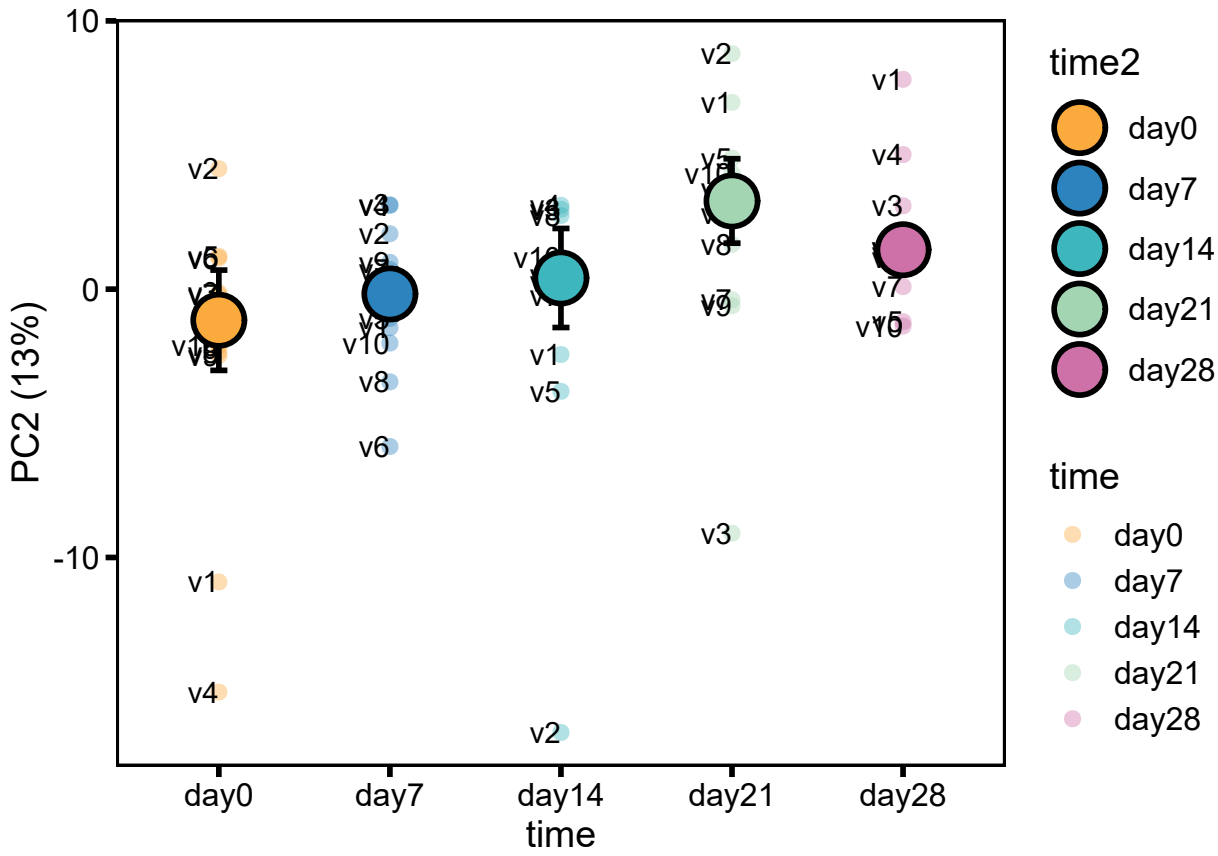

```
# boxplot with median summarisation PC1
pca <- doPCA(RNAseq$filt$DESeq_vst_nobatch)

df <- pca$pcs %>%
  cbind(RNAseq$filt$design)

df2 <- df %>%
  group_by(time) %>%
  select(time, PC1) %>%
  summarize_each(dplyr::funs(median, sd, se=sd(.)/sqrt(n())), PC1) %>%
  dplyr::rename(time2 = time)

ggplot() +
  geom_point(data = df, aes(x=time, y=PC1, color=time), shape=16, size=3, stroke=0, alpha=0.4) +
  geom_text(data = df,aes(x=time,y=PC1,label=volunteer),hjust=1)+
  geom_errorbar(data=df2, aes(x=time2, y=median, ymin=median-se, ymax=median+se), width=.1, lwd=1) +
  geom_point(data = df2, aes(x=time2, y=median, fill=time2), color="black", shape=21, size=8, stroke=1.5, alpha=1) +
  theme_custom(legend.position = "right") +
  ylab(paste("PC1 (", round(pca$percentVar[1],0), "%)", sep = "")) +
  scale_color_manual(values = colPals$time) +
  scale_fill_manual(values = colPals$time)
```



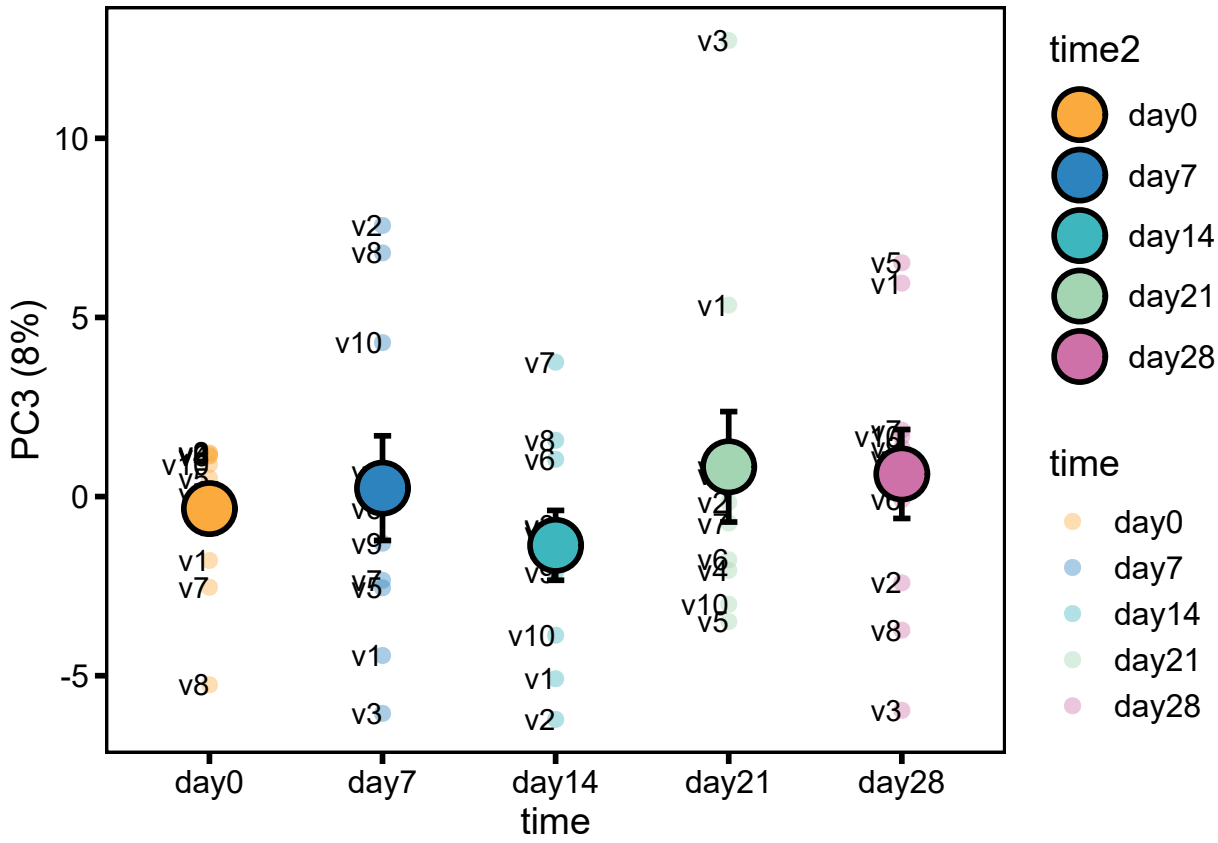

```
#boxplot with mean summarisation PC2
pca <- doPCA(RNAseq$filt$DESeq_vst_nobatch)

df <- pca$pcs %>%
  cbind(RNAseq$filt$design)

df2 <- df %>%
  group_by(time) %>%
  select(time, PC2) %>%
  summarize_each(dplyr::funs(mean, sd, se=sd(.)/sqrt(n())), PC2) %>%
  dplyr::rename(time2 = time)

ggplot() +
  geom_point(data = df, aes(x=time, y=PC2, color=time), shape=16, size=3, stroke=0, alpha=0.4) +
  geom_text(data = df,aes(x=time,y=PC2,label=volunteer),hjust=1)+
  geom_errorbar(data=df2, aes(x=time2, y=mean, ymin=mean-se, ymax=mean+se), width=.1, lwd=1) +
  geom_point(data = df2, aes(x=time2, y=mean, fill=time2), color="black", shape=21, size=8, stroke=1.5, alpha=1) +
  theme_custom(legend.position = "right") +
  ylab(paste("PC2 (", round(pca$percentVar[2],0), "%)", sep = "")) +
  scale_color_manual(values = colPals$time) +
  scale_fill_manual(values = colPals$time)
```



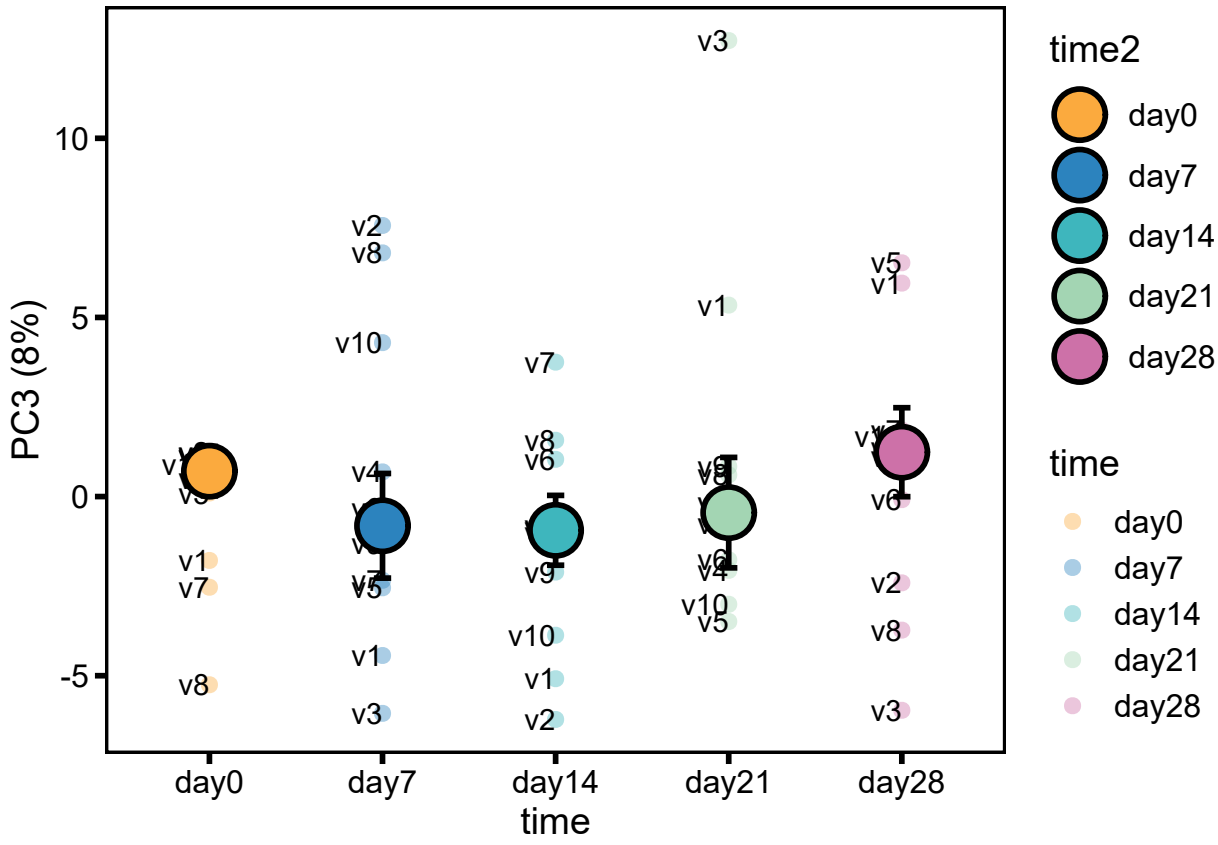

```
#boxplot with median summarisation PC2
pca <- doPCA(RNAseq$filt$DESeq_vst_nobatch)

df <- pca$pcs %>%
  cbind(RNAseq$filt$design)

df2 <- df %>%
  group_by(time) %>%
  select(time, PC2) %>%
  summarize_each(dplyr::funs(median, sd, se=sd(.)/sqrt(n())), PC2) %>%
  dplyr::rename(time2 = time)

ggplot() +
  geom_point(data = df, aes(x=time, y=PC2, color=time), shape=16, size=3, stroke=0, alpha=0.4) +
  geom_text(data = df,aes(x=time,y=PC2,label=volunteer),hjust=1)+
  geom_errorbar(data=df2, aes(x=time2, y=median, ymin=median-se, ymax=median+se), width=.1, lwd=1) +
  geom_point(data = df2, aes(x=time2, y=median, fill=time2), color="black", shape=21, size=8, stroke=1.5, alpha=1) +
  theme_custom(legend.position = "right") +
  ylab(paste("PC2 (", round(pca$percentVar[2],0), "%)", sep = "")) +
  scale_color_manual(values = colPals$time) +
  scale_fill_manual(values = colPals$time)
```




```
#boxplot with mean summarisation PC3
pca <- doPCA(RNAseq$filt$DESeq_vst_nobatch)

df <- pca$pcs %>%
  cbind(RNAseq$filt$design)

df2 <- df %>%
  group_by(time) %>%
  select(time, PC3) %>%
  summarize_each(dplyr::funs(mean, sd, se=sd(.)/sqrt(n())), PC3) %>%
  dplyr::rename(time2 = time)

ggplot() +
  geom_point(data = df, aes(x=time, y=PC3, color=time), shape=16, size=3, stroke=0, alpha=0.4) +
  geom_text(data = df,aes(x=time,y=PC3,label=volunteer),hjust=1)+
  geom_errorbar(data=df2, aes(x=time2, y=mean, ymin=mean-se, ymax=mean+se), width=.1, lwd=1) +
  geom_point(data = df2, aes(x=time2, y=mean, fill=time2), color="black", shape=21, size=8, stroke=1.5, alpha=1) +
  theme_custom(legend.position = "right") +
  ylab(paste("PC3 (", round(pca$percentVar[3],0), "%)", sep = "")) +
  scale_color_manual(values = colPals$time) +
  scale_fill_manual(values = colPals$time)
```




```
#boxplot with median summarisation PC3
pca <- doPCA(RNAseq$filt$DESeq_vst_nobatch)

df <- pca$pcs %>%
  cbind(RNAseq$filt$design)

df2 <- df %>%
  group_by(time) %>%
  select(time, PC3) %>%
  summarize_each(dplyr::funs(median, sd, se=sd(.)/sqrt(n())), PC3) %>%
  dplyr::rename(time2 = time)

ggplot() +
  geom_point(data = df, aes(x=time, y=PC3, color=time), shape=16, size=3, stroke=0, alpha=0.4) +
  geom_text(data = df,aes(x=time,y=PC3,label=volunteer),hjust=1)+
  geom_errorbar(data=df2, aes(x=time2, y=median, ymin=median-se, ymax=median+se), width=.1, lwd=1) +
  geom_point(data = df2, aes(x=time2, y=median, fill=time2), color="black", shape=21, size=8, stroke=1.5, alpha=1) +
  theme_custom(legend.position = "right") +
  ylab(paste("PC3 (", round(pca$percentVar[3],0), "%)", sep = "")) +
  scale_color_manual(values = colPals$time) +
  scale_fill_manual(values = colPals$time)
```

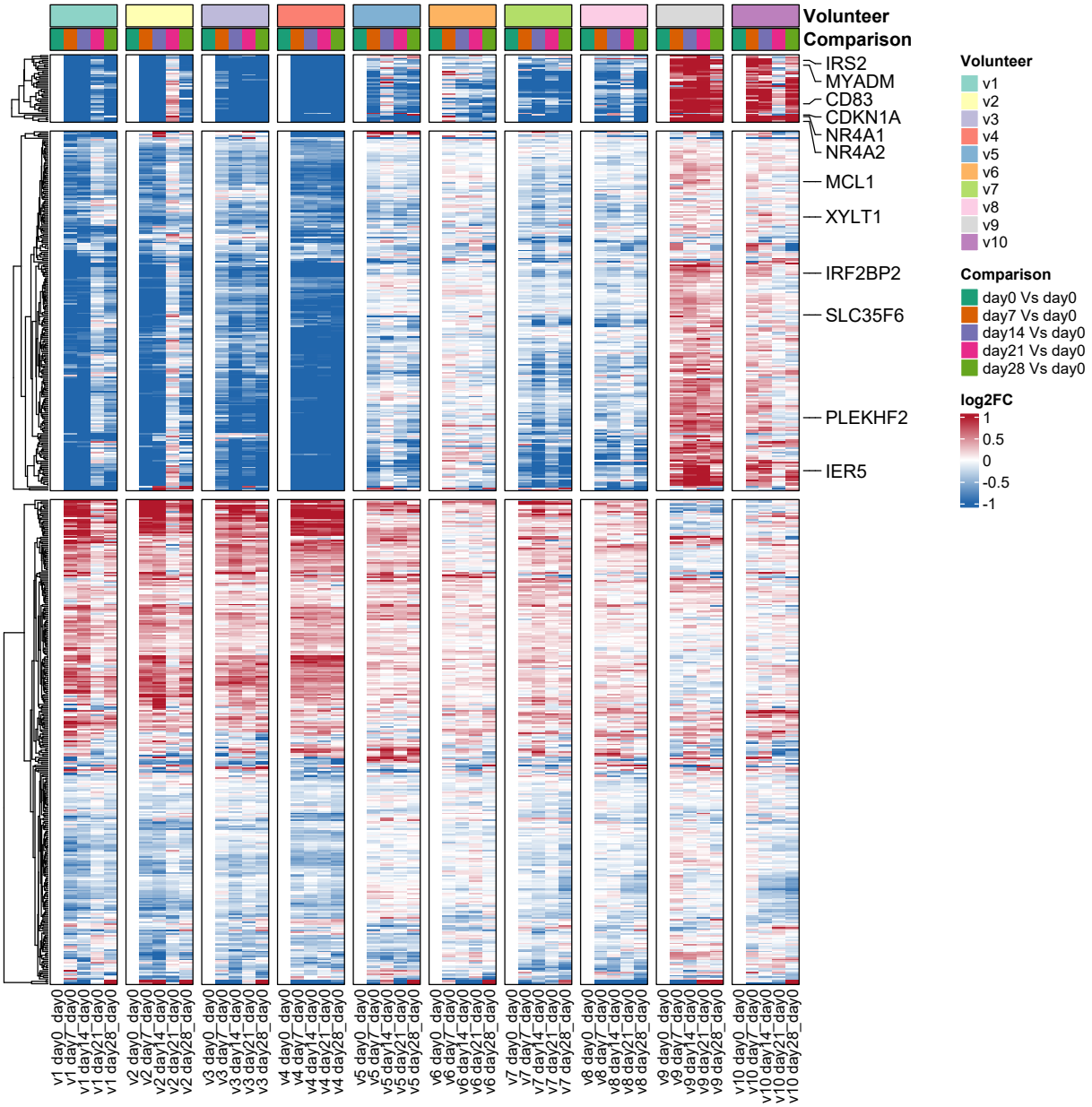


Heatmap of DE gene fold change per volunteer

```
##  [1] "v1"  "v2"  "v3"  "v4"  "v5"  "v6"  "v7"  "v8"  "v9"  "v10"
## [1] "v1"
## [1] "v2"
## [1] "v3"
## [1] "v4"
## [1] "v5"
## [1] "v6"
## [1] "v7"
## [1] "v8"
## [1] "v9"
## [1] "v10"
```

```
mark.genes <- c('CDKN1A','NR4A1','NR4A2','MYADM','IRS2', 'CD83',
                'MCL1','XYLT1','IRF2BP2','PLEKHF2','IER5','SLC35F6')

m <- FCOutBg[rownames(FCOutBg) %in% DESeq2_DEGs_filt$padj_02$GLMtime$GeneSymbol,]

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
)

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"))
)

hc_tree <- hclust(dist(m, method = 'euclidean'), method = 'complete')
clust <- cutree(hc_tree, k=3)

p <- Heatmap(m, name = "log2FC",
             row_split=factor(clust, levels = c(3,2,1)), 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(-1, 1, length.out=21),
                                        colors=colorRampPalette(c("#2166AC", "white", "#B2182B"))(21)),
             top_annotation = ha_top,
             right_annotation = ha_right,
             width = unit(170, "mm"),
             show_row_names = F, 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")
```



```
pdf("plots/figS3_heatmap_GLMtime_DEGs_fdr02_all_volunteers.pdf", width = 10, height = 10)
draw(p, merge_legend = T, align_heatmap_legend = "heatmap_top")
dev.off()
```

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

```
# add cluster information
df <- m %>%
  as.data.frame() %>%
  rownames_to_column(var = 'GeneSymbol') %>%
  add_column(Cluster = recode(.$GeneSymbol, !!!clust), .after = 'GeneSymbol') %>%
  arrange(desc(Cluster))
colnames(df) <- gsub(' ', '_', colnames(df))

head(df)
```

```
##   GeneSymbol Cluster v1_day0_day0 v1_day7_day0 v1_day14_day0 v1_day21_day0
## 1    SNORD55       3            0    -2.555755     -3.100324    -2.2474142
## 2    GADD45A       3            0    -3.498003     -3.849928    -0.3460918
## 3     TENT5C       3            0    -5.941867     -4.415470    -0.6892786
## 4       LMNA       3            0    -3.341684     -3.122239    -1.7824649
## 5       RGS2       3            0    -2.891241     -3.197108    -1.3121941
## 6       BTG2       3            0    -2.677152     -3.203831    -0.2567009
##   v1_day28_day0 v2_day0_day0 v2_day7_day0 v2_day14_day0 v2_day21_day0
## 1    -1.0596099            0    -2.361799     -4.288352    0.54818444
## 2    -1.4872851            0    -3.881534     -4.078899    0.64532812
## 3    -1.5360862            0    -4.124591     -4.863170   -0.03369748
## 4    -1.9272633            0    -3.163733     -2.900545   -0.54052050
## 5    -1.8856238            0    -3.094777     -1.531801    0.05688757
## 6    -0.1475172            0    -3.244029     -3.054314    0.27180629
##   v2_day28_day0 v3_day0_day0 v3_day7_day0 v3_day14_day0 v3_day21_day0
## 1    -3.4971052            0   -4.2115255     -4.666718     -3.704865
## 2    -1.3191874            0   -0.9945114     -3.641905     -2.138811
## 3    -1.4524291            0   -1.6222933     -3.801901     -2.385167
## 4    -1.4564632            0   -2.4730788     -3.244047     -1.511365
## 5    -0.6457841            0   -0.8666366     -2.985528     -1.143766
## 6    -0.3284787            0   -0.6583088     -2.652504     -1.454440
##   v3_day28_day0 v4_day0_day0 v4_day7_day0 v4_day14_day0 v4_day21_day0
## 1     -4.159354            0    -4.638925    -0.4668353    -0.2943534
## 2     -2.530564            0    -4.762237    -4.8350593    -5.0677225
## 3     -4.677311            0    -5.434495    -5.3686521    -5.7826529
## 4     -2.451281            0    -4.194071    -3.7918571    -3.7018251
## 5     -2.037847            0    -3.508136    -3.3399718    -3.5262364
## 6     -1.892690            0    -2.572266    -2.7798637    -3.0511374
##   v4_day28_day0 v5_day0_day0 v5_day7_day0 v5_day14_day0 v5_day21_day0
## 1     -5.096689            0    0.9406646     0.2269535     1.6110676
## 2     -5.353752            0   -0.8049867     0.3798050    -0.3164543
## 3     -5.733997            0   -0.8953110     0.5832916    -0.7666483
## 4     -3.712691            0   -0.6821185    -0.3026732    -1.0363464
## 5     -4.608578            0   -0.6665507     0.6666530    -0.3400228
## 6     -4.139334            0   -1.3979919    -0.2881216    -0.8464381
##   v5_day28_day0 v6_day0_day0 v6_day7_day0 v6_day14_day0 v6_day21_day0
## 1    0.08364401            0   -1.1113216    -1.9952208    -1.1637307
## 2   -0.75569910            0   -0.7216933    -0.6353398    -0.9964948
## 3   -0.60395551            0   -0.2705872    -0.4928175    -0.5601680
## 4    0.26137441            0   -0.3722137    -0.4844233    -0.2769652
## 5   -0.47418236            0    0.1557997     0.2824366    -0.3330260
## 6   -1.14626092            0    0.6041653    -0.3396026    -0.6406625
##   v6_day28_day0 v7_day0_day0 v7_day7_day0 v7_day14_day0 v7_day21_day0
## 1  -0.257496605            0    0.3226219    -1.6390420    -1.4882147
## 2  -0.770642757            0   -0.5551318    -0.8629478    -0.1099975
## 3  -0.682106816            0   -1.0730738    -0.9967463    -0.8703376
## 4  -0.001913474            0   -1.0953689    -0.7192124    -1.4160649
## 5  -0.321396078            0   -0.4984369    -2.0649872    -0.5909377
## 6  -0.479241690            0   -1.7743363    -3.1194326    -1.5599699
##   v7_day28_day0 v8_day0_day0 v8_day7_day0 v8_day14_day0 v8_day21_day0
## 1     -2.286330            0   -1.5910509     0.4758088   -0.10690660
## 2     -1.478975            0   -0.4434937    -0.8269000   -0.21910868
## 3     -1.264850            0    0.4848386    -0.9347691   -0.04364865
## 4     -1.105758            0    0.7388150     0.9827688    0.89128324
## 5     -2.031201            0   -0.9325318    -1.3502999   -0.31276779
## 6     -2.313138            0   -1.2774450    -1.4533614   -0.28814125
##   v8_day28_day0 v9_day0_day0 v9_day7_day0 v9_day14_day0 v9_day21_day0
## 1    -1.8172074            0   -1.2302058    -0.4438439    -0.3225177
## 2    -1.1309107            0    1.4634409     0.9988098     1.8814505
## 3    -0.4436283            0    1.4542367     1.7726102     2.0297752
## 4     0.9255282            0    0.7952818     2.1825208     2.6578440
## 5    -1.1566376            0    0.7150694     0.5704871     0.9795493
## 6    -0.9755160            0    0.5932704     1.2218654     1.2541658
##   v9_day28_day0 v10_day0_day0 v10_day7_day0 v10_day14_day0 v10_day21_day0
```

```
## 1    0.14059077             0    0.04597312     -0.5915053    -0.40348078
## 2    1.18047827             0    1.34028150      0.6660343     0.27391168
## 3    0.79671625             0    1.36796374      0.8252115    -0.53587054
## 4    0.08485952             0    1.60149823      1.0380204    -0.24140983
## 5    0.50719739             0    0.66820496      0.3004350    -0.16144176
## 6    0.76099277             0    0.49181900      0.9057045    -0.06929183
##   v10_day28_day0
## 1     -0.4564566
## 2      2.0459528
## 3      0.8452243
## 4      0.4277745
## 5      0.8635131
## 6      0.8805784
```

Exports

```
# DE gene clusters all volunteers
openxlsx::write.xlsx(df, file = "tables/dataS1_GLMtime_DEGs_fdr02_all_volunteers.xlsx", rowNames=F, overwrite=T)

# RNAseq count table excluding volunteers 9 & 10
# df <- read.table(file = 'data/rnaseq/exon_counts_all.txt',stringsAsFactors = FALSE,sep = "\t",header = TRUE)
# df <- df[,!grepl("v10",colnames(df)) & !grepl("v9",colnames(df))]
# write.table(df, file = 'data/rnaseq/rnaseq_count_mtx.tsv', quote = F, sep = '\t', col.names = T, row.names = F)

# RNAseq design table excluding volunteers 9 & 10
# df <- RNAseq$unfilt$design[!RNAseq$unfilt$design$volunteer %in% c("v9","v10"),]
# write.table(df, file = 'data/rnaseq/rnaseq_design_mtx.tsv', quote = F, sep = '\t', col.names = T, row.names = F)
```

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      stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] ComplexHeatmap_2.14.0       RColorBrewer_1.1-3
##  [3] variancePartition_1.28.9    BiocParallel_1.32.6
##  [5] limma_3.54.2                DESeq2_1.38.3
##  [7] SummarizedExperiment_1.28.0 Biobase_2.58.0
##  [9] MatrixGenerics_1.10.0       matrixStats_0.63.0
## [11] GenomicRanges_1.50.2        GenomeInfoDb_1.34.9
## [13] IRanges_2.32.0              S4Vectors_0.36.2
## [15] BiocGenerics_0.44.0         patchwork_1.1.2
## [17] magrittr_2.0.3              forcats_1.0.0
## [19] stringr_1.5.0               dplyr_1.1.1
## [21] purrr_1.0.1                 readr_2.1.4
## [23] tidyr_1.3.0                 tibble_3.2.1
## [25] ggplot2_3.4.2               tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
##   [1] readxl_1.4.2            backports_1.4.1         circlize_0.4.15
##   [4] plyr_1.8.8              remaCor_0.0.11          splines_4.2.1
##   [7] digest_0.6.31           foreach_1.5.2           htmltools_0.5.5
##  [10] fansi_1.0.4             memoise_2.0.1           googlesheets4_1.1.0
##  [13] cluster_2.1.3           doParallel_1.0.17       aod_1.3.2
##  [16] openxlsx_4.2.5.1        tzdb_0.3.0              Biostrings_2.66.0
##  [19] annotate_1.76.0         modelr_0.1.11           timechange_0.2.0
##  [22] prettyunits_1.1.1       colorspace_2.1-0        ggrepel_0.9.3
##  [25] blob_1.2.4              rvest_1.0.3             haven_2.5.2
##  [28] rbibutils_2.2.13        xfun_0.38               crayon_1.5.2
```

```
##  [31] RCurl_1.98-1.12         jsonlite_1.8.4          lme4_1.1-32
##  [34] iterators_1.0.14        glue_1.6.2              gtable_0.3.3
##  [37] gargle_1.3.0            zlibbioc_1.44.0         XVector_0.38.0
##  [40] GetoptLong_1.0.5        DelayedArray_0.24.0     shape_1.4.6
##  [43] scales_1.2.1            mvtnorm_1.1-3           DBI_1.1.3
##  [46] Rcpp_1.0.10             xtable_1.8-4            progress_1.2.2
##  [49] clue_0.3-64             bit_4.0.5               httr_1.4.5
##  [52] gplots_3.1.3            pkgconfig_2.0.3         XML_3.99-0.14
##  [55] farver_2.1.1            dbplyr_2.3.2            locfit_1.5-9.7
##  [58] utf8_1.2.3              tidyselect_1.2.0        labeling_0.4.2
##  [61] rlang_1.1.0             reshape2_1.4.4          AnnotationDbi_1.60.2
##  [64] munsell_0.5.0           cellranger_1.1.0        tools_4.2.1
##  [67] cachem_1.0.7            cli_3.6.1               generics_0.1.3
##  [70] RSQLite_2.3.0           broom_1.0.4             evaluate_0.20
##  [73] fastmap_1.1.1           yaml_2.3.7              RhpcBLASctl_0.23-42
##  [76] knitr_1.42              bit64_4.0.5             fs_1.6.1
##  [79] zip_2.2.2               caTools_1.18.2          KEGGREST_1.38.0
##  [82] nlme_3.1-157            xml2_1.3.3              compiler_4.2.1
##  [85] pbkrtest_0.5.2          rstudioapi_0.14         png_0.1-8
##  [88] reprex_2.0.2            clusterGeneration_1.3.7 geneplotter_1.76.0
##  [91] stringi_1.7.12          highr_0.10              lattice_0.20-45
##  [94] Matrix_1.5-3            nloptr_2.0.3            vctrs_0.6.1
##  [97] pillar_1.9.0            lifecycle_1.0.3         RUnit_0.4.32
## [100] Rdpack_2.4              GlobalOptions_0.1.2     bitops_1.0-7
## [103] R6_2.5.1                KernSmooth_2.23-20      codetools_0.2-18
## [106] boot_1.3-28             MASS_7.3-57             gtools_3.9.4
## [109] rjson_0.2.21            withr_2.5.0             GenomeInfoDbData_1.2.9
## [112] parallel_4.2.1          hms_1.1.3               minqa_1.2.5
## [115] rmarkdown_2.21          googledrive_2.1.0       lubridate_1.9.2
```