basic

```r
library(data.table)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(stringr)
library(ggplot2)
library(NMF)
```

```
## Loading required package: pkgmaker

## Loading required package: registry

##
## Attaching package: 'pkgmaker'

## The following object is masked from 'package:base':
##
##     isNamespaceLoaded

## Loading required package: rngtools

## Loading required package: cluster

## NMF - BioConductor layer [NO: missing Biobase] | Shared memory capabilities [NO: bigmemory] | Cores

##   To enable the Bioconductor layer, try: install.extras('
## NMF
## ') [with Bioconductor repository enabled]
##   To enable shared memory capabilities, try: install.extras('
## NMF
## ')
```

```r
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

```r
library(parallel)
library(RColorBrewer)
library(scales)
```

```r
load("mixcr_processed.Rda")
dt.clones <- dt.clones %>% filter(replica == 1)
#dt.clones <- dt.clones %>% filter(replica == 2 | group == "control")
```

V usage

```r
dt.vusage = dt.clones %>%
#  mutate(freq = freq/10) %>% #to normalize frequencies after 10 time sampling
  group_by(sample_name, vaccination, age, v) %>%
  summarise(freq = sum(freq)) %>% filter(freq > 0.001)

mat.vusage = dt.vusage %>%
  dcast(sample_name + vaccination + age ~ v, fill = 0)
```

```
## Using freq as value column: use value.var to override.
```

```r
dt.annot = mat.vusage[,2:3]

rownames(mat.vusage) = mat.vusage$sample_name
mat.vusage = mat.vusage[,4:ncol(mat.vusage)]

ann_colors = c()

i = 2
for (annR in colnames(dt.annot)) {
  n = length(unique(dt.annot[[annR]]))
  tmp = list(x = c(brewer.pal(n, name = paste0("Set",i))[1:n]))
  names(tmp) = annR
  ann_colors = c(ann_colors, tmp)
  i = i + 1
}
```

```
## Warning in brewer.pal(n, name = paste0("Set", i)): minimal value for n is 3, returning requested pal
```

```r
pdf("figures/p1.pdf", width = 6, height = 8)
aheatmap(pmin(as.matrix(t(mat.vusage)), 0.15),
         hclustfun = "ward",
         annCol = dt.annot, annColors = ann_colors,
         scale = "none")
```

```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

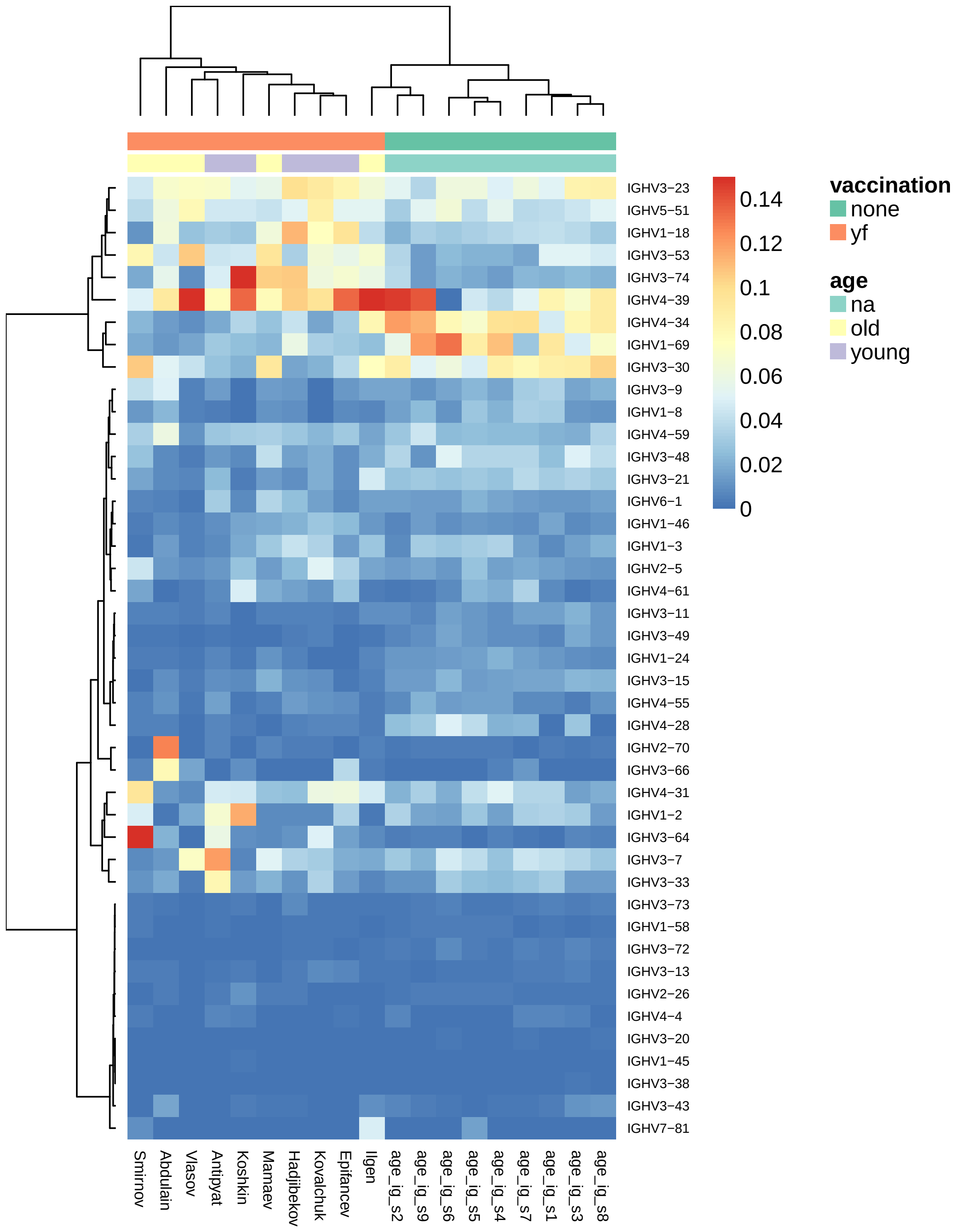
```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

```r
dev.off()
```

```
## pdf
##   2
```

```r
aheatmap(pmin(as.matrix(t(mat.vusage)), 0.15),
         hclustfun = "ward",
         annCol = dt.annot, annColors = ann_colors,
         scale = "none")
```

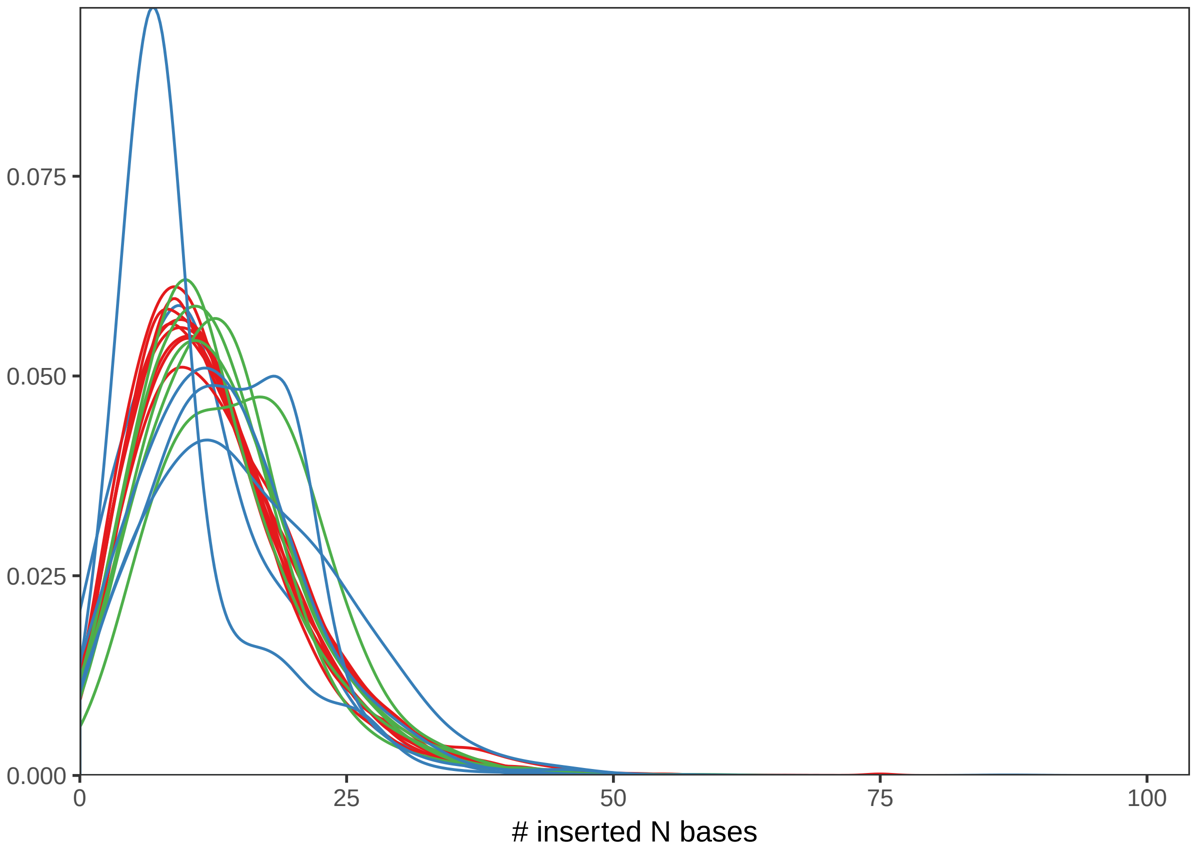
```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```



VDJ junction structure

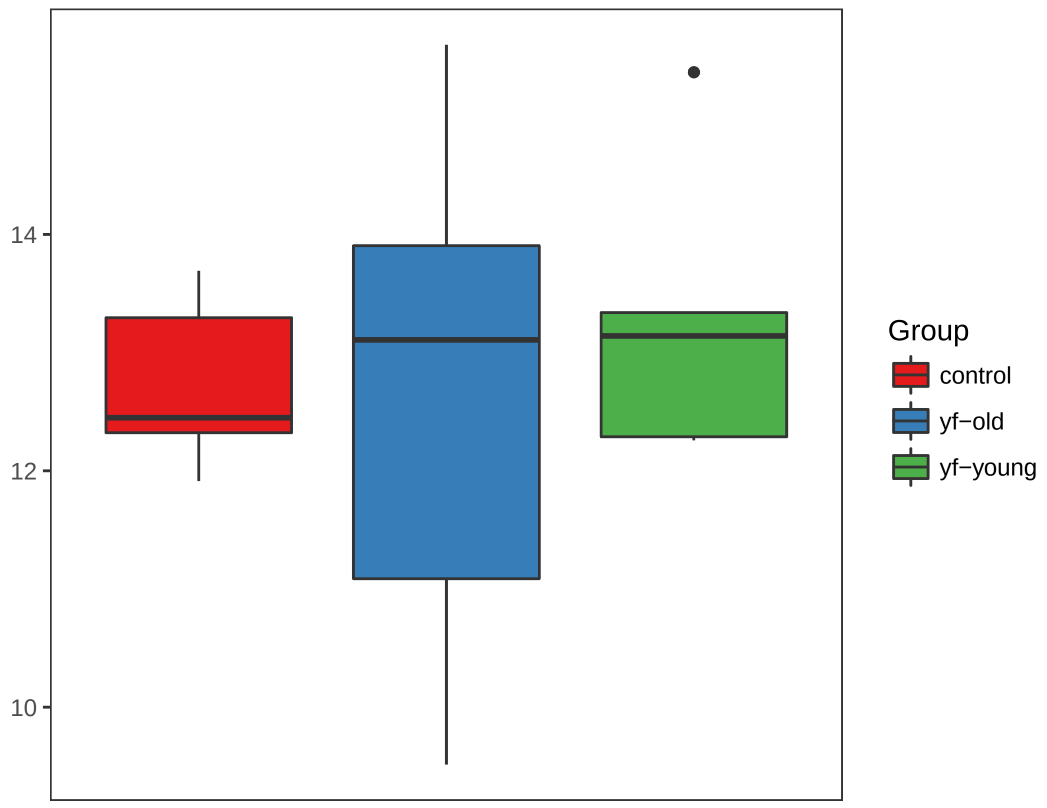
```
dt.ins = dt.clones %>% filter(!is.na(dEnd)) %>%
  mutate(nIns = dStart - vEnd + jStart - dEnd) %>%
  group_by(sample_name) %>%
  mutate(freq = freq/sum(freq))

p2a=ggplot(dt.ins, aes(x = nIns, group = sample_name, weight = freq, color = group)) +
  geom_density(adjust = 2) +
  #scale_x_continuous(limits=c(-0.001,50)) +
  scale_color_brewer(guide = F, palette = "Set1") +
  scale_x_continuous("# inserted N bases", expand = c(0,0)) + scale_y_continuous("", expand = c(0,0)) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
p2a
```



```
dt.ins.s = dt.ins %>%
  group_by(group, sample_name) %>%
  summarise(nInsS = sum(nIns * freq))

p2b=ggplot(dt.ins.s, aes(x = group, group = group, y = nInsS, fill = group)) +
  geom_boxplot() +
  #scale_x_continuous(limits=c(-0.001,50)) +
  scale_fill_brewer("Group", palette = "Set1") +
  xlab("") + ylab("") +
  theme_bw() +
  theme(aspect=1,
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        axis.text.x = element_blank(), axis.ticks.x = element_blank())
p2b
```



```
a=aov(nInsS~group,dt.ins.s)
summary(a)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## group        2   1.33  0.6661   0.333  0.722
## Residuals   16  32.00  2.0003
```

```
TukeyHSD(a, "group")
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = nInsS ~ group, data = dt.ins.s)
##
## $group
##                          diff       lwr      upr     p adj
## yf-old-control    -0.05734436 -2.092877 1.978188 0.9970915
## yf-young-control   0.57847472 -1.457058 2.614007 0.7476991
## yf-young-yf-old    0.63581908 -1.672258 2.943896 0.7607005
```

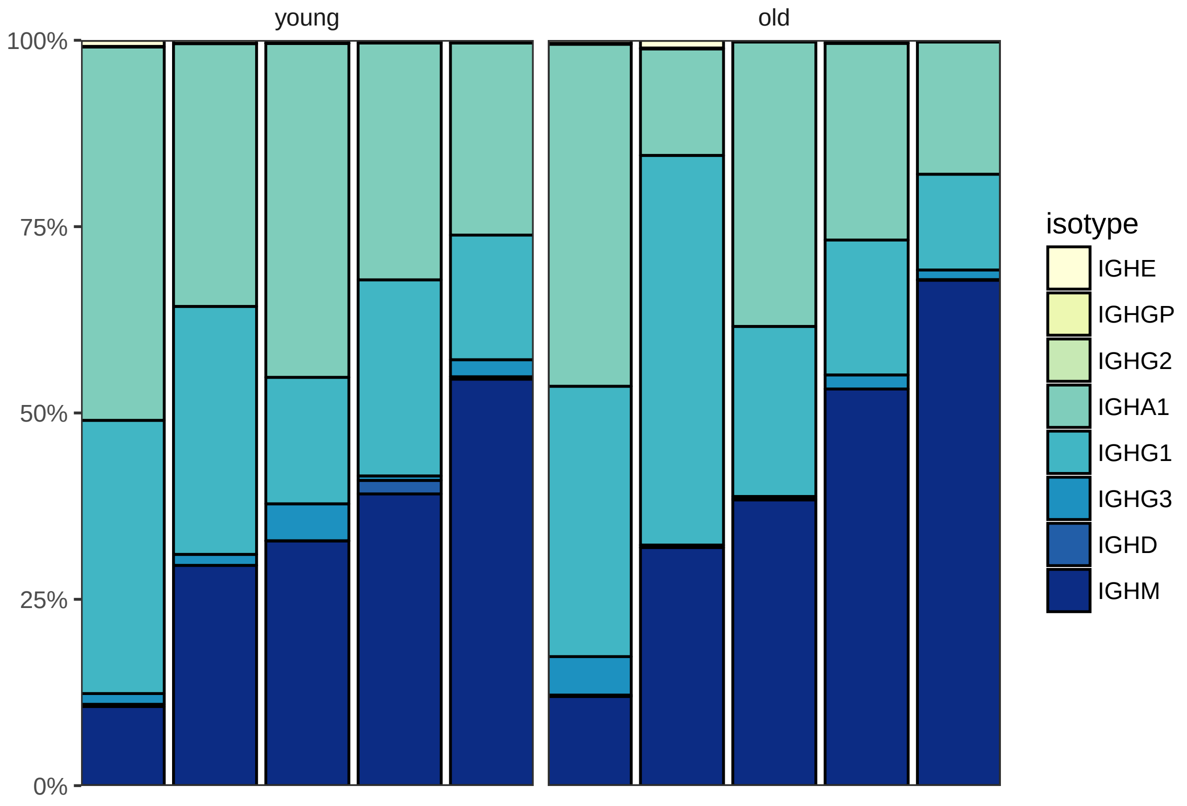
Isotype usage

```
dt.iso = dt.clones %>%
  filter(vaccination != "none", isotype != "", !is.na(isotype)) %>%
  group_by(sample_name, age, isotype) %>%
  summarise(freq = sum(freq)) %>%
  group_by(sample_name) %>%
  mutate(freq = freq / sum(freq))
```

```
dt.iso$sample_name = factor(dt.iso$sample_name,
                            with(dt.iso %>% filter(isotype == "IGHM"), sample_name[order(freq)]))

dt.iso$isotype = factor(dt.iso$isotype, levels = rev(c("IGHM", "IGHD", "IGHG3", "IGHG1", "IGHA1", "IGHG
dt.iso$age = factor(dt.iso$age, levels = c('young', 'old'))
dt.iso <- filter(dt.iso, !is.na(isotype))
```

```
p3=ggplot(dt.iso, aes(x=sample_name, fill = isotype, y = freq)) +
  geom_bar(position = "stack", stat = "identity", color = "black") +
  facet_wrap(~age, scales = "free_x") +
  scale_x_discrete("",expand = c(0,0)) +
  scale_y_continuous("", expand = c(0,0),labels = percent) +
  scale_fill_brewer(palette = "YlGnBu") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        axis.text.x = element_blank(), axis.ticks.x = element_blank(),
        strip.background = element_blank())
p3
```



```
dt.p = data.table()
for (i in unique(dt.iso$isotype)) {
  a = aov(freq~age, dt.iso %>% filter(isotype == i))
  p = summary(a)[[1]][["Pr(>F)"]][1]
  dt.p = rbind(dt.p,
               data.table(isotype = i, p.adj = p))
}
dt.p$p.adj = p.adjust(dt.p$p.adj, method="BH")
print(dt.p)
```

```
##    isotype     p.adj
## 1:   IGHA1 0.8471394
## 2:    IGHE 0.9307657
## 3:   IGHG1 0.8804539
## 4:   IGHG3 0.8804539
## 5:   IGHGP 0.8471394
## 6:    IGHM 0.8804539
## 7:    IGHD 0.8471394
## 8:   IGHG2 0.8471394
```

Amino acid feature analysis

```
dt.clones.cdr3prop = dt.clones %>%
  group_by(sample_name, group, cdr3aa) %>%
  summarise(freq = sum(freq))
```

```
dt.cdr3.flat = unique(dt.clones.cdr3prop$cdr3aa) %>%
  strsplit("") %>%
  mclapply(function(x) data.table(aa = x, cdr3aa = paste0(x, collapse = "")), mc.cores = 60) %>%
  rbindlist
```

```
dt.cdr3.flat.ann = dt.cdr3.flat %>%
  merge(fread("kidera.txt") %>% mutate(Len = 1) %>% melt, allow.cartesian = T) %>%
  group_by(cdr3aa, variable) %>%
  summarise(value = sum(value))
```

```
## Using aa as id variables
```

```
dt.clones.cdr3prop2 = dt.clones.cdr3prop %>%
  merge(dt.cdr3.flat.ann, by = "cdr3aa", allow.cartesian = T)
```

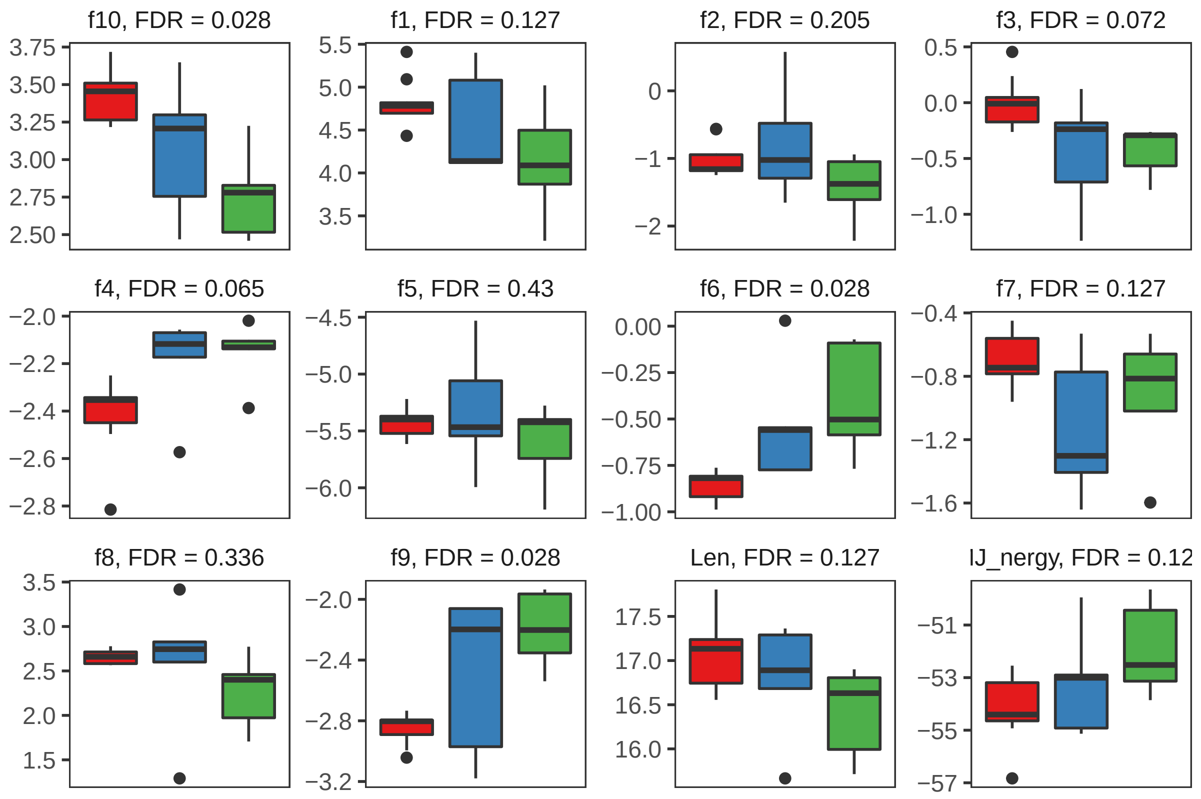
```
dt.clones.cdr3prop.s = dt.clones.cdr3prop2 %>%
  group_by(sample_name, group, variable) %>%
  summarise(value = sum(value #/ ifelse(variable == "Len", 1, nchar(cdr3aa))
                      * freq))
```

```
dt.p = data.table()
for (v in unique(dt.clones.cdr3prop.s$variable)) {
  a = aov(value~group,dt.clones.cdr3prop.s %>% filter(variable == v))
  p = summary(a)[[1]][["Pr(>F)"]][1]
  dt.p = rbind(dt.p,
               data.table(variable = v, p.adj = p))
}
dt.p$p.adj = p.adjust(dt.p$p.adj, method="BH")

dt.clones.cdr3prop.s1 <- dt.clones.cdr3prop.s %>%
         merge(dt.p) %>%
         mutate(variable2 = paste0(variable, ", FDR = ", round(p.adj, 3)))

#dt.clones.cdr3prop.s1$variable2 = factor(dt.clones.cdr3prop.s1$variable2, levels = c("f1, FDR = 0.127"
#                                                                                   "f4, FDR = 0.065"
#                                                                                   "f7, FDR = 0", "f
#                                                                                   "f10, FDR = 0", "

p4=ggplot(dt.clones.cdr3prop.s1,
      aes(x = group, fill = group, y = value)) +
  geom_boxplot() +
  scale_fill_brewer(guide = F, palette = "Set1") +
  facet_wrap(~variable2, scales = "free_y") + xlab("") + ylab("") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        axis.text.x = element_blank(), axis.ticks.x = element_blank(),
        strip.background = element_blank())
p4
```



```
ggsave("figures/p2a.pdf", p2a, width = 8, height = 4)
ggsave("figures/p2b.pdf", p2b, width = 4, height = 4)
ggsave("figures/p3.pdf", p3, width = 8, height = 4)
ggsave("figures/p4.pdf", p4, width = 4*2, height = 3*2)
```