graph analysis

```
library(ggplot2)
library(stringr)
library(reshape2)
library(dplyr)
```

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

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

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

```
library(ggbeeswarm)
library(ineq)
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following objects are masked from 'package:reshape2':
##
##     dcast, melt
```

```
select = dplyr::select
summarise = dplyr::summarise

load('clones_rep12.rda')

gs = mutate_each(clones, funs(as.double(.)), -root, -cdr3aa, -sample, -single) %>%
  mutate(cdr3aa = as.character(cdr3aa),
         proj = ifelse(sample %in% c("Abdulain", "Ilgen", "Mamaev", "Smirnov", "Vlasov"), 'old', 'young
```

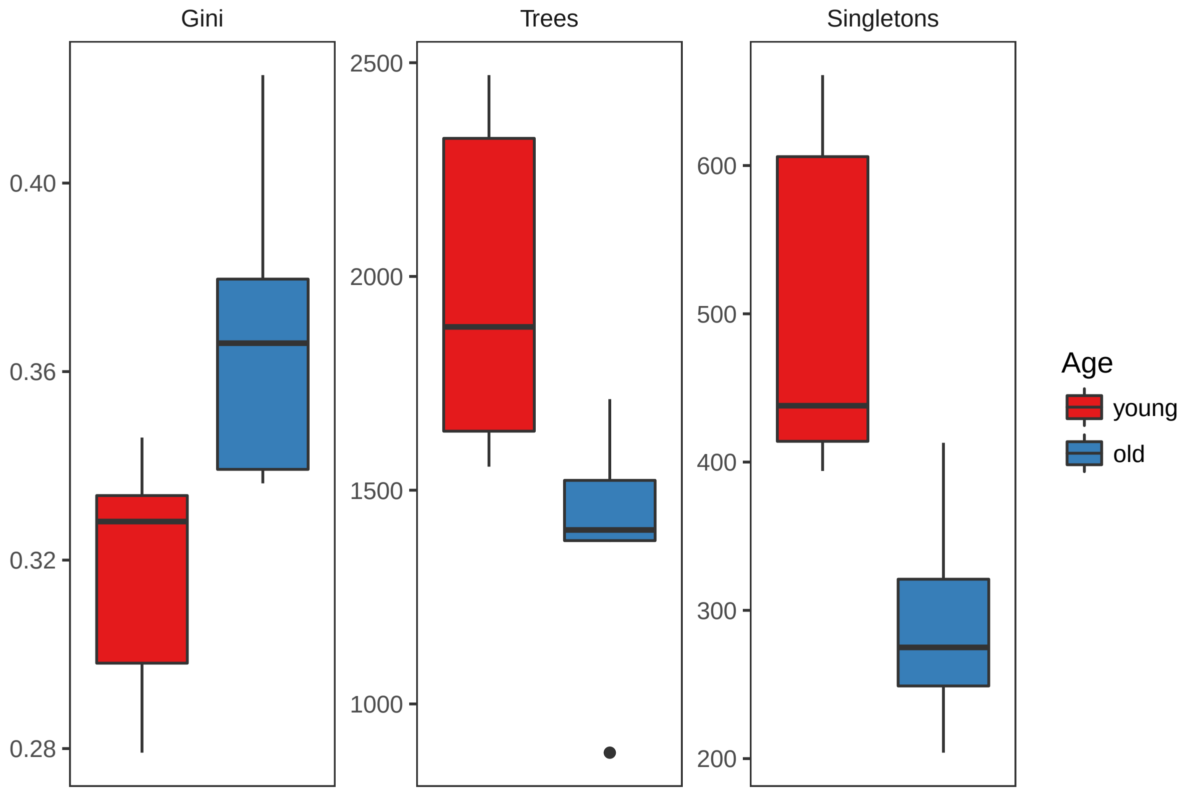
```
## `mutate_each()` is deprecated.
## Use `mutate_all()`, `mutate_at()` or `mutate_if()` instead.
## To map `funs` over a selection of variables, use `mutate_at()`

## Warning in evalq(as.double(proj), <environment>): NAs introduced by
## coercion
```

```
gs$proj = factor(gs$proj, levels = c('young', 'old'))
```

```
gs.s = gs %>%
  group_by(proj, sample) %>%
  summarise(Gini = ineq(nodes, type='Gini'),
            Trees = n(), Singletons = sum(single))

gs.s2 = gs.s %>% melt
```

```
## Using proj, sample as id variables
```

```
p11=ggplot(gs.s2, aes(x = proj, fill=proj, y = value)) +
  geom_boxplot() +
  facet_wrap(~variable, scales = "free") +
  ylab("") + xlab("") +
  scale_fill_brewer("Age", palette = "Set1") +
  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())
p11
```

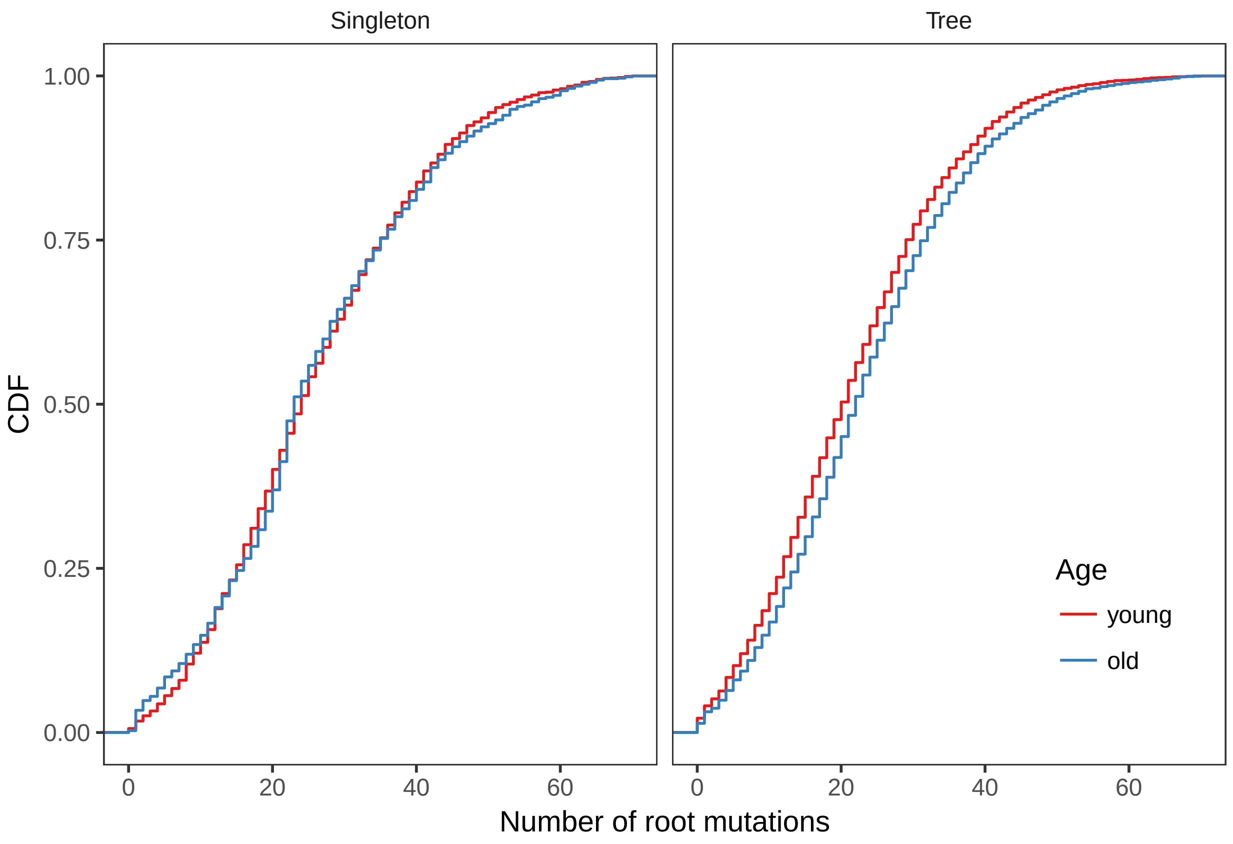


```
dt.p = data.table()

for (vv in unique(gs.s2$variable)) {
  tt = t.test(value ~ proj, gs.s2 %>% filter(variable == vv))
  p = tt$p.value
  dt.p = rbind(dt.p,
               data.table(variable = vv, p=p))
}

dt.p$p.adj = p.adjust(dt.p$p, method = "BH")
print(dt.p %>% arrange(p.adj))
```

```
##     variable          p      p.adj
## 1       Gini 0.03385023 0.03406631
## 2      Trees 0.03406631 0.03406631
## 3 Singletons 0.01489365 0.03406631
```

```r
p12=ggplot(gs %>% mutate(single = ifelse(single, "Singleton", "Tree")), aes(x = root.mut, color = proj))
  stat_ecdf() +
  facet_wrap(~single) +
  scale_color_brewer("Age", palette = "Set1") +
  scale_x_continuous("Number of root mutations", limits = c(0,70)) + ylab("CDF") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        strip.background = element_blank(),
        legend.position = c(0.90, 0.2))
p12
```

```
## Warning: Removed 158 rows containing non-finite values (stat_ecdf).
```



```r
ks.test(gs %>% filter(single == T, proj == "old") %>% .$root.mut,
        gs %>% filter(single == T, proj != "old") %>% .$root.mut)
```

```
## Warning in ks.test(gs %>% filter(single == T, proj == "old") %>% .
## $root.mut, : p-value will be approximate in the presence of ties

##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  gs %>% filter(single == T, proj == "old") %>% .$root.mut and gs %>% filter(single == T, proj
## D = 0.036334, p-value = 0.1741
## alternative hypothesis: two-sided
```

```r
ks.test(gs %>% filter(single == F, proj == "old") %>% .$root.mut,
        gs %>% filter(single == F, proj != "old") %>% .$root.mut)
```

```
## Warning in ks.test(gs %>% filter(single == F, proj == "old") %>% .
## $root.mut, : p-value will be approximate in the presence of ties

##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  gs %>% filter(single == F, proj == "old") %>% .$root.mut and gs %>% filter(single == F, proj
## D = 0.065308, p-value = 5.066e-12
## alternative hypothesis: two-sided
```

```r
ggsave("figures/p11.pdf", p11, width=6,height=4)
ggsave("figures/p12.pdf", p12, width=6,height=4)
```

```
## Warning: Removed 158 rows containing non-finite values (stat_ecdf).
```