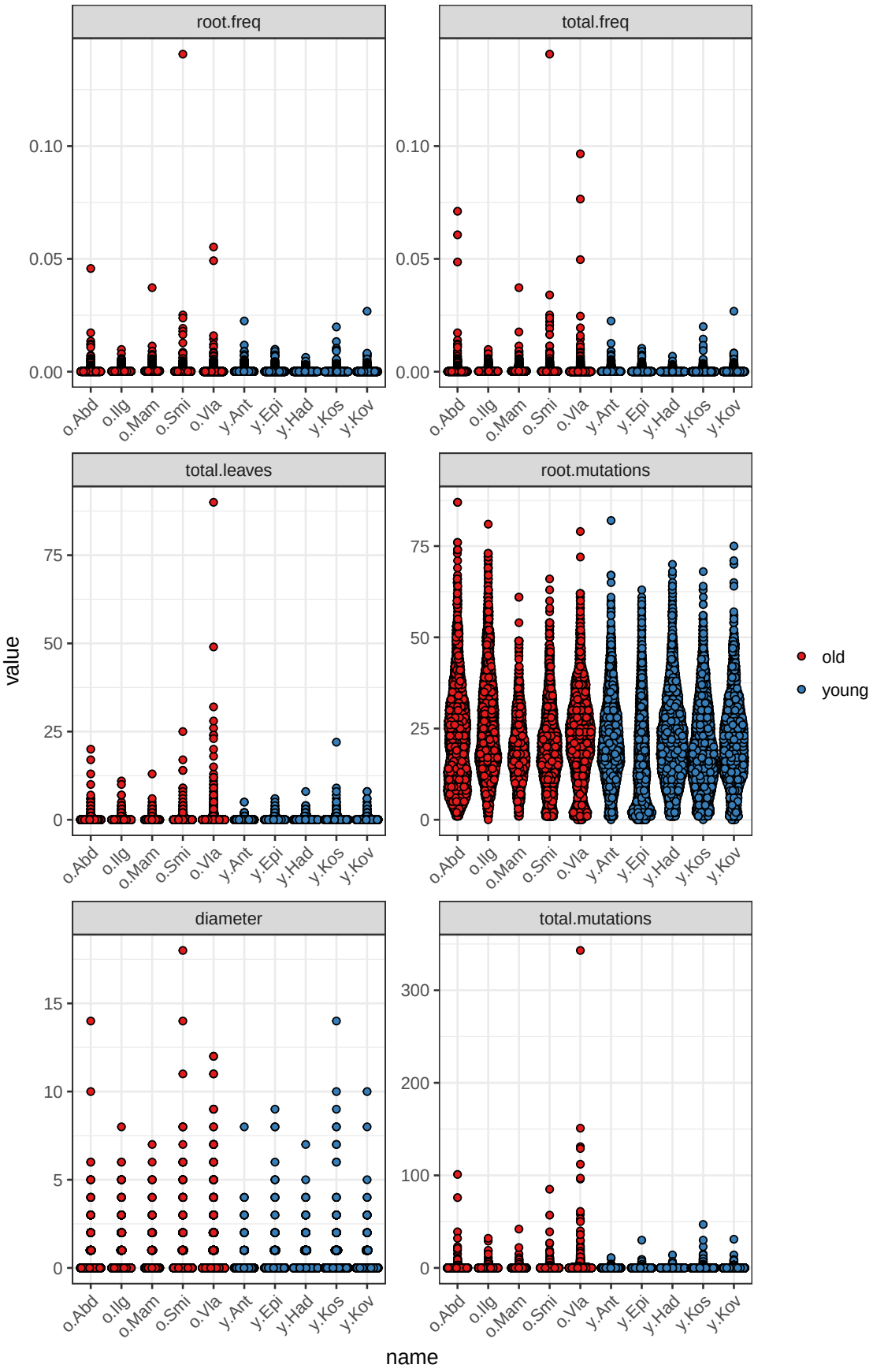
Phylogenetic analysis of clonal trees

Anna Obraztsova

2/16/2017

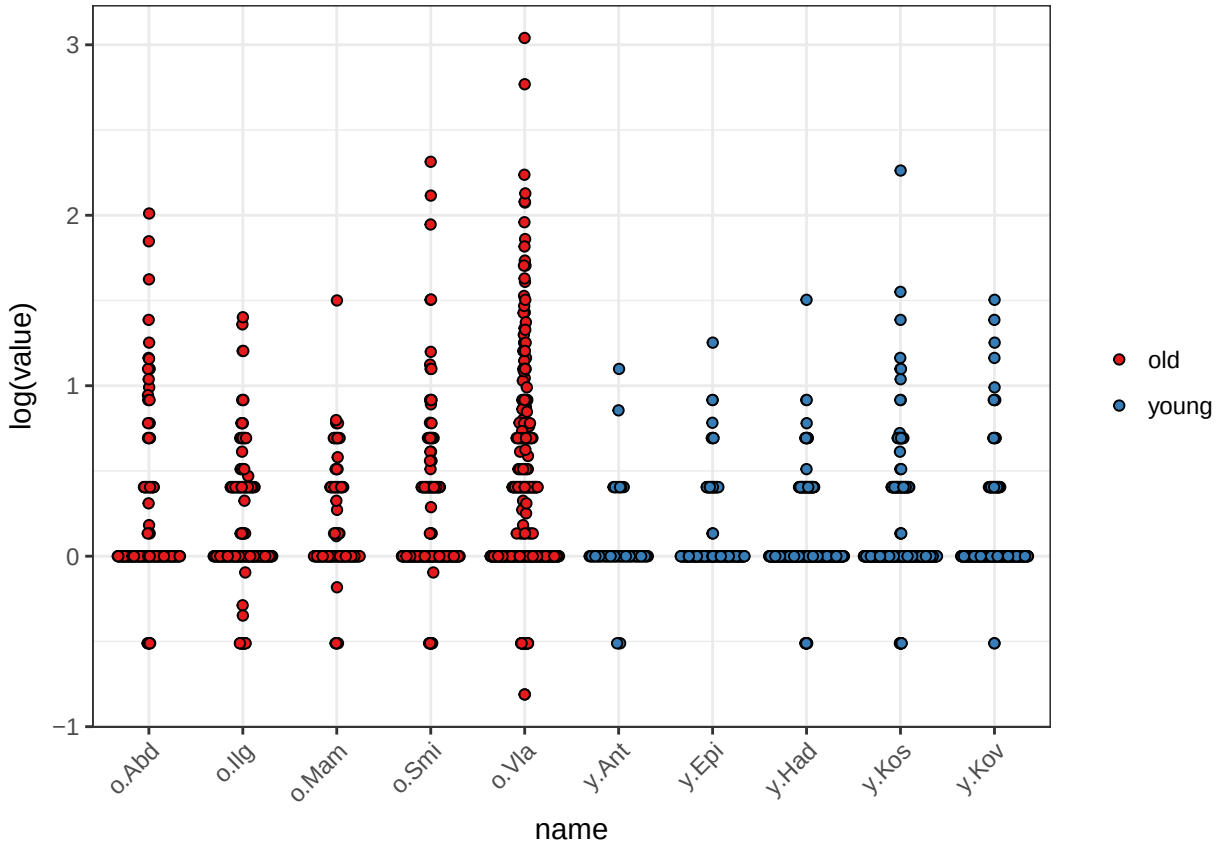
Diameter is the largest number of mutations between root and leave. Branching is a number of leaves divided by mean length of path from root to leave. Total.freq is a sum of frequencies of all vertices of tree.

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

old_rna = c("Abdulain", "Ilgen", "Mamaev", "Smirnov", "Vlasov")
young_rna = c("Antipyat", "Epifancev", "Hadjibekov", "Koshkin", "Kovalchuk")

gs <- data.frame()

for (sample in old_rna){
  .df <- read.table(paste('~/yf/trees/stat/yf_old_RNA/', sample, ".txt", sep = ""), header=T, sep="\t")
  names(.df) <- c('ndn', 'root.freq', 'v', 'j', 'total.freq', 'total.leaves', 'total.nodes', 'root.muta
    .df$proj <- "old"
    .df$sample <- sample
    gs <- rbind(gs, .df)
}

for (sample in young_rna){
    .df <- read.table(paste('~/yf/trees/stat/yf_young_RNA/', sample, ".txt", sep = ""), header=T, sep="
    names(.df) <- c('ndn', 'root.freq', 'v', 'j', 'total.freq', 'total.leaves', 'total.nodes', 'root.mu
    .df$proj <- "young"
    .df$sample <- sample
    gs <- rbind(gs, .df)
}

gs <- mutate_each(gs, funs(as.double(.)), -ndn, -proj, -sample, -v, -j)

gs2 <- gs %>% dplyr::select(-j, -v) %>%
  melt(id=c('ndn', 'proj', 'sample')) %>%
  mutate(name=paste(str_sub(proj, 1, 1), str_sub(sample,1, 3), sep='.')) %>%
  filter(variable != 'total.nodes' & variable != 'mean.path')

first = c('root.freq', 'total.freq', 'total.leaves', 'root.mutations', 'diameter', 'total.mutations')
ggplot(filter(gs2, variable %in% first)) + geom_quasirandom(aes(x = name, y = value, group=proj, fill =
facet_wrap(~variable, nrow=4, scales='free') +
scale_fill_brewer("", palette = "Set1") +
theme_bw() +
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


root.freq
total.freq
total.leaves
root.mutations
diameter
total.mutations
value
name
old
young
0.00
0.05
0.10
0
25
50
75
0
5
10
15
0
100
200
300
o.Abd
o.Ilg
o.Mam
o.Smi
o.Vla
y.Ant
y.Epi
y.Had
y.Kos
y.Kov

```
ggplot(filter(gs2, variable == 'branching')) + geom_quasirandom(aes(x = name, y = log(value), group=proj
scale_fill_brewer("", palette = "Set1") +
theme_bw() +
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Removed 16401 rows containing missing values (geom_point).
```

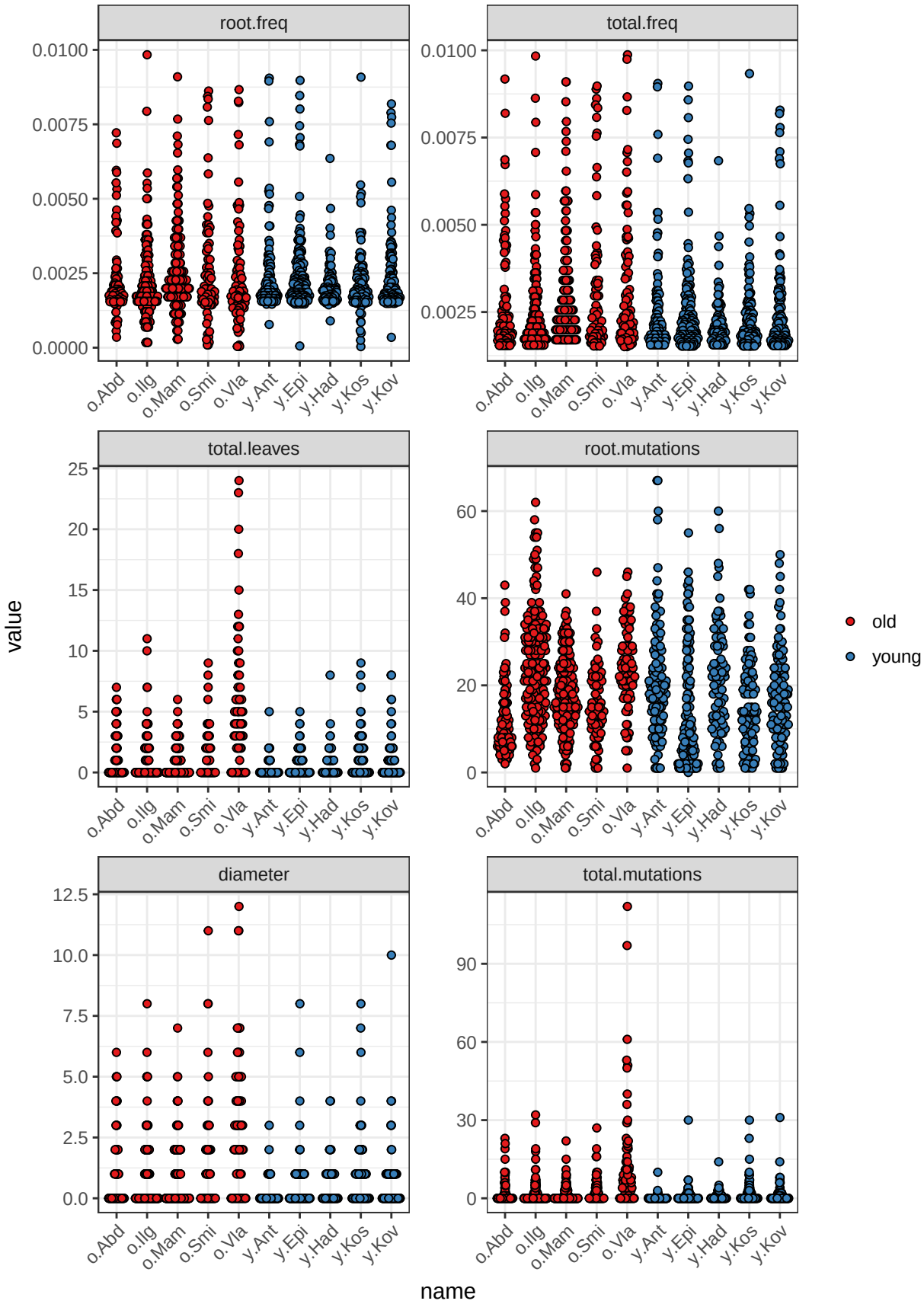



```
gs2.1 <- gs %>% filter(total.freq > 0.0015 & total.freq < 0.01) %>%
  melt(id=c('ndn', 'proj', 'sample')) %>%
  mutate(name=paste(str_sub(proj, 1, 1), str_sub(sample,1, 3), sep='.'), value = as.double(value)) %>%
  filter(variable != 'total.nodes' & variable != 'mean.path')
```

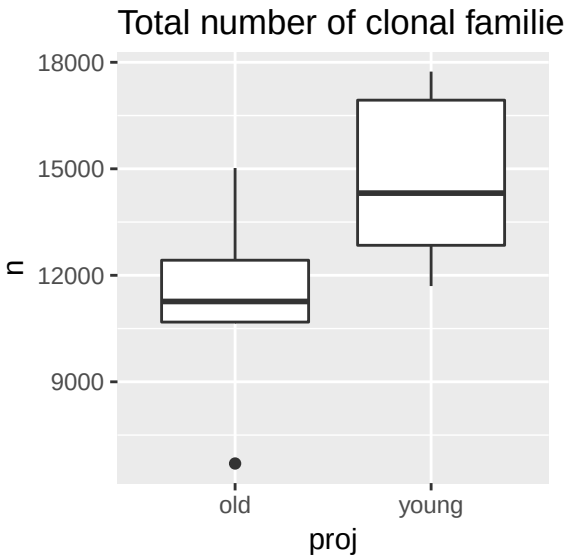
```
## Warning: attributes are not identical across measure variables; they will
## be dropped
```

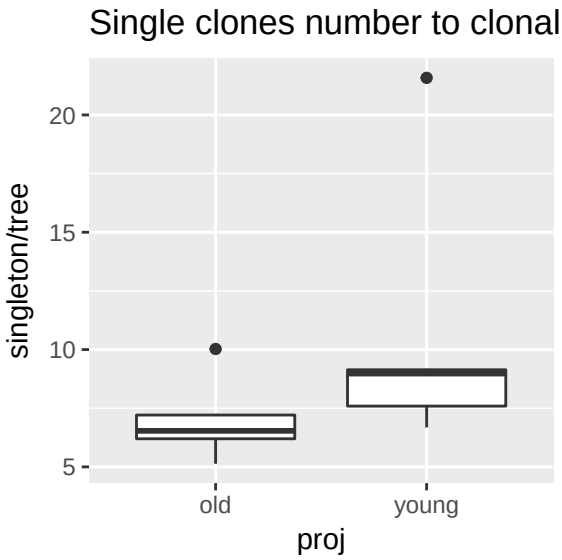
```
## Warning in eval(substitute(expr), envir, enclos): NAs introduced by
## coercion
```

```
ggplot(filter(gs2.1, variable %in% first)) + geom_quasirandom(aes(x = name, y = value, group=proj, fill
facet_wrap(~variable, nrow=4, scales='free') +
scale_fill_brewer("", palette = "Set1") +
theme_bw() +
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

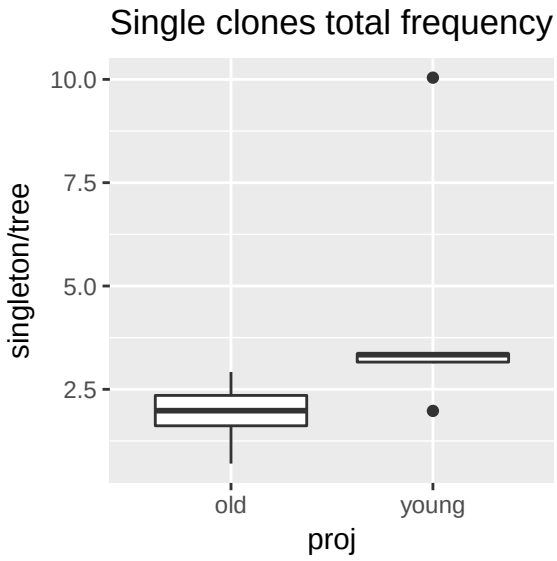

root.freq
total.freq
total.leaves
root.mutations
diameter
total.mutations
value
name
old
young
o.Abd
o.Ilg
o.Mam
o.Smi
o.Vla
y.Ant
y.Epi
y.Had
y.Kos
y.Kov

```
gs3 <- gs2 %>% dplyr::group_by(sample, proj) %>% summarise(n = n())
ggplot(gs3) + geom_boxplot(aes(x = proj, y = n)) + ggtitle('Total number of clonal families')

gs4 <- gs %>% mutate(single = ifelse(total.leaves == 0, 'singleton', 'tree')) %>%
  dplyr::group_by(sample, proj, single) %>% summarise(n = n()) %>%
  dcast(sample + proj ~ single)
ggplot(gs4, aes(x=proj, y = singleton/tree)) + geom_boxplot() + ggtitle('Single clones number to clonal
```

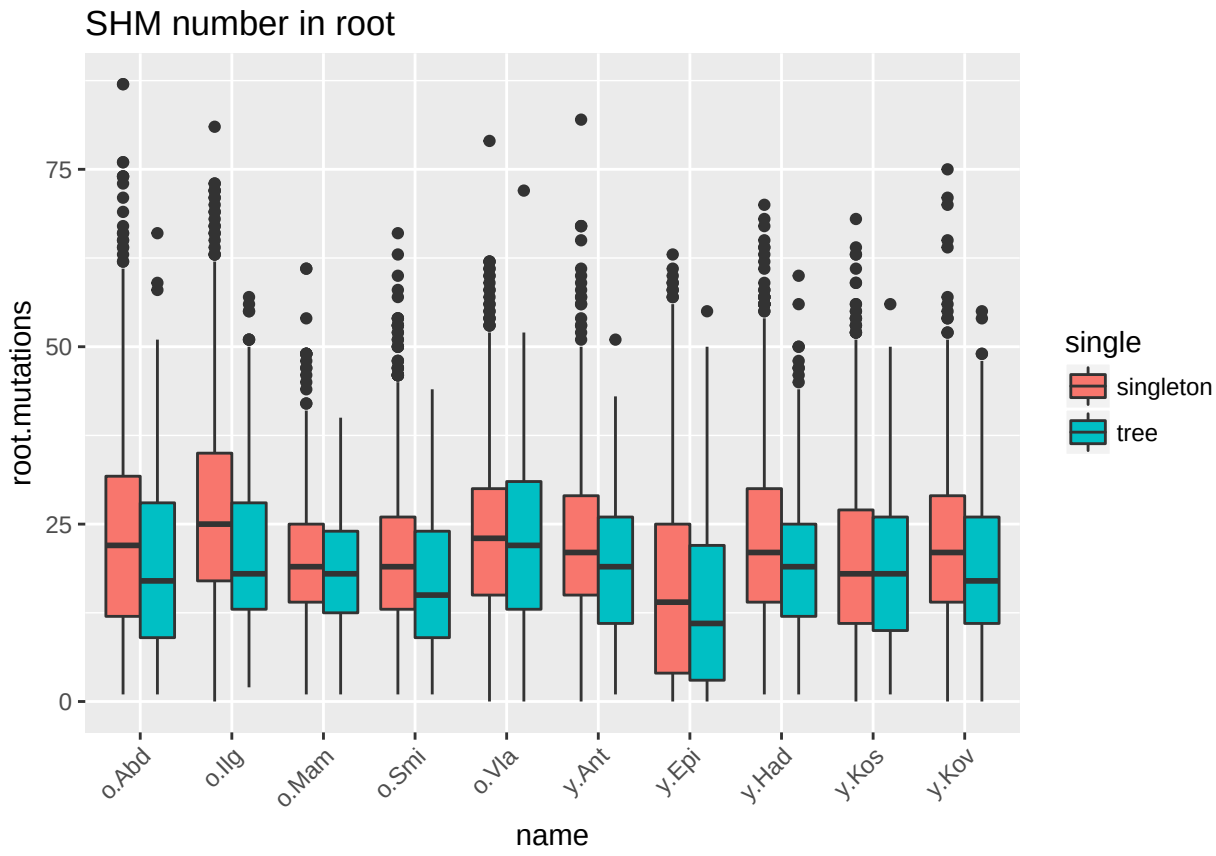




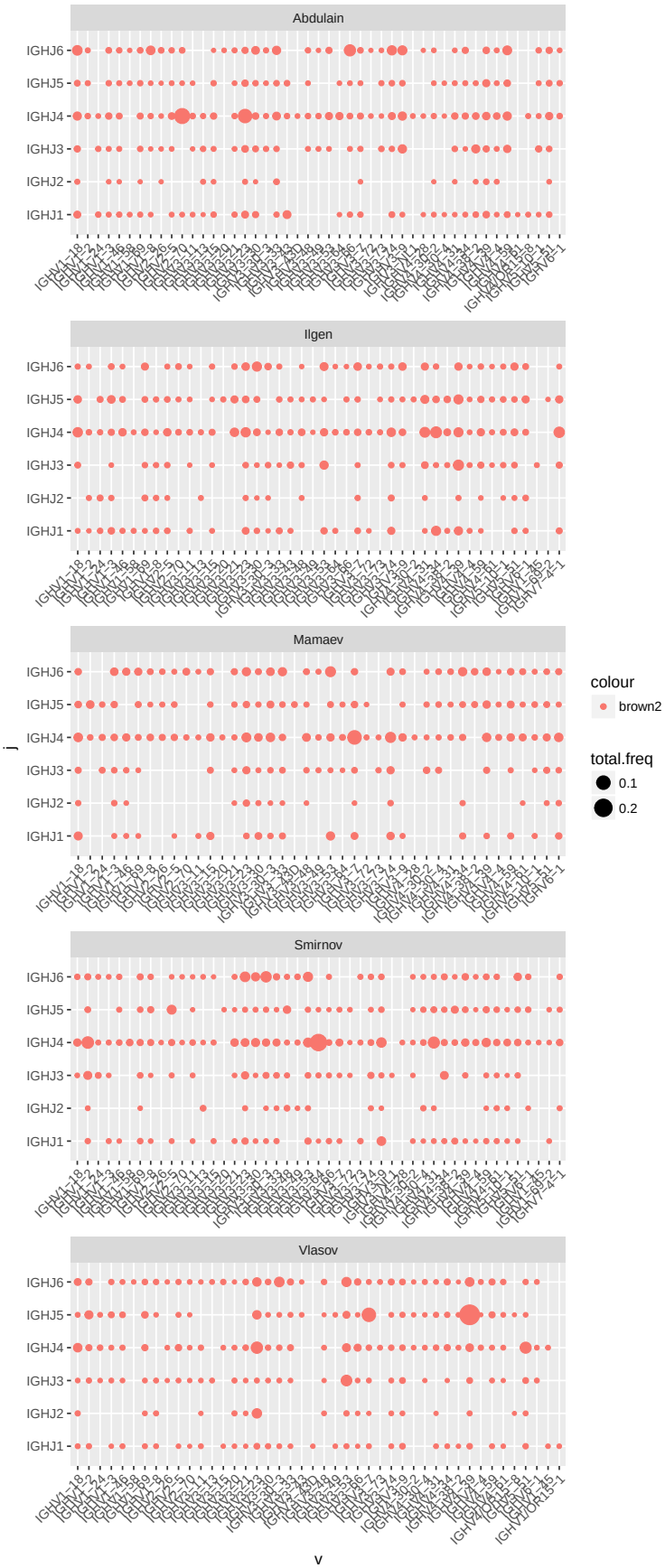

```
gs4.1 <- gs %>% mutate(single = ifelse(total.leaves == 0, 'singleton', 'tree')) %>%
  dplyr::group_by(sample, proj, single) %>% summarise(n = sum(total.freq)) %>%
  dcast(sample + proj ~ single)
ggplot(gs4.1, aes(x=proj, y = singleton/tree)) + geom_boxplot() + ggtitle('Single clones total frequency
```



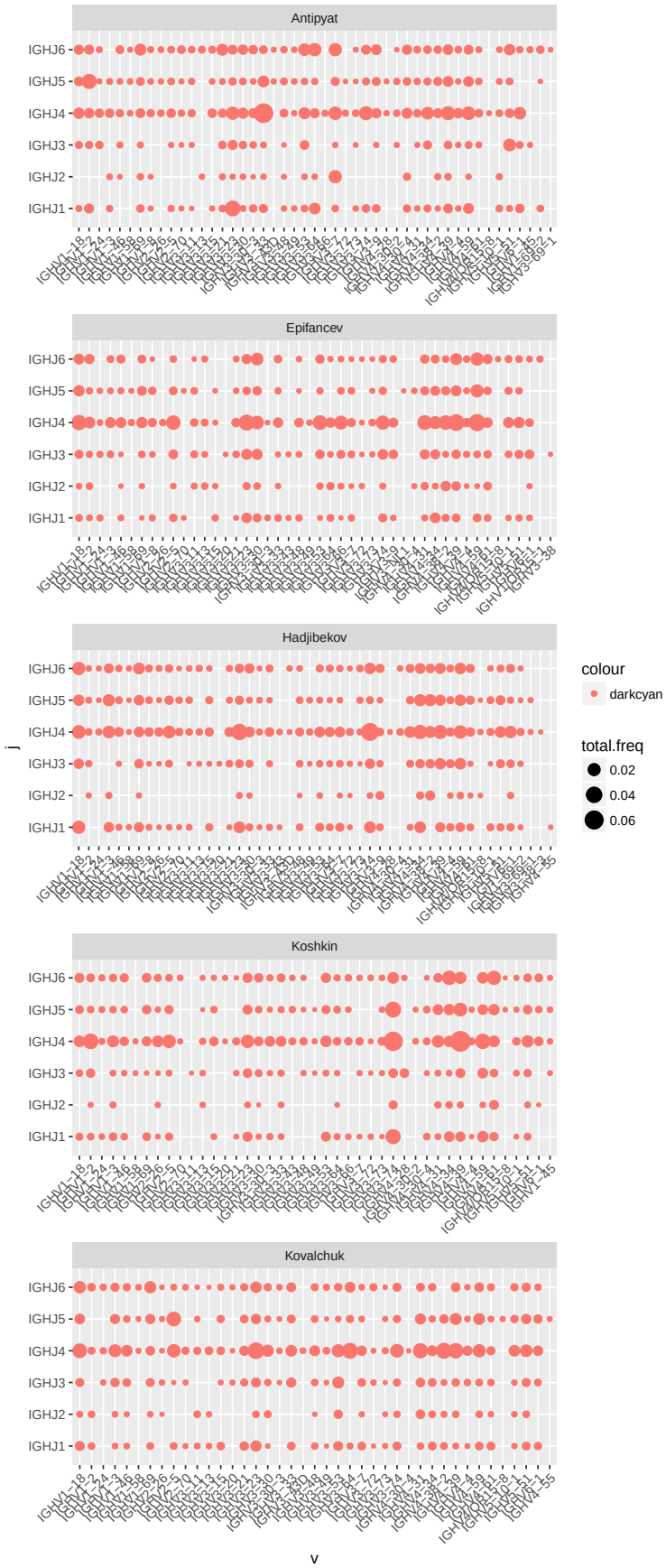

```
gs4.2 <- gs %>% mutate(single = ifelse(total.leaves == 0, 'singleton', 'tree')) %>%
  mutate(name=paste(str_sub(proj, 1, 1), str_sub(sample,1, 3), sep='.'))
ggplot(gs4.2, aes(x = name, y = root.mutations, group=interaction(sample, single), fill=single)) + geom
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
gs5 <- gs %>% dplyr::group_by(sample, proj, v, j) %>% summarise(total.freq = sum(total.freq))

ggplot(filter(gs5, proj=='old'), aes(x = v, y = j, size=total.freq, col='brown2')) + geom_point() +
facet_wrap(~sample, ncol=1, scales='free') +
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Abdulain
Ilgen
Mamaev
Smirnov
Vlasov
IGHJ6
IGHJ5
IGHJ4
IGHJ3
IGHJ2
IGHJ1
j
v
colour
brown2
total.freq
0.1
0.2

```
ggplot(filter(gs5, proj=='young'), aes(x = v, y = j, size=total.freq, col='darkcyan')) + geom_point() +
facet_wrap(~sample, ncol=1, scales='free') +
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Antipyat
Epifancev
Hadjibekov
Koshkin
Kovalchuk
IGHJ6
IGHJ5
IGHJ4
IGHJ3
IGHJ2
IGHJ1
j
v
colour
darkcyan
total.freq
0.02
0.04
0.06