

Distribution, RS ratio and patterns of novel SHM

Mikhail Shugay, Anna Obraztsova

17/04/2017

Analysis of substitution type and frequency

Load preprocessed data

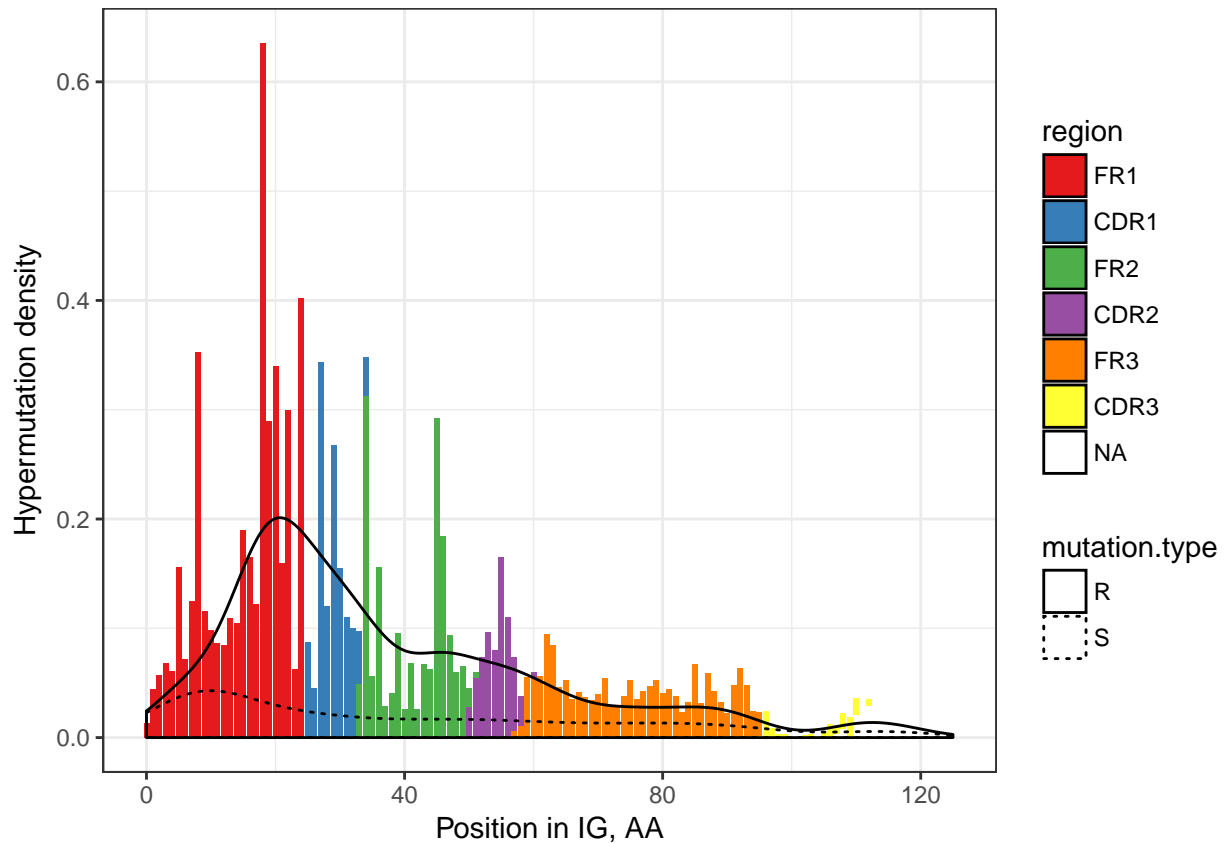
```
library(plyr)
library(ggplot2)
load("sp_new.Rda")
df <- subset(df, type == "RNA")
df$region <- factor(df$region, c("FR1", "CDR1", "FR2", "CDR2", "FR3", "CDR3"))
df$mutation.type <- ifelse(df$from.aa == df$to.aa, "S", "R")
df <- ddply(df, .(proj, type, sample, replica), transform,
            weight = total.clonotypes / sum(total.clonotypes))
```

Check if we observe well-documented increase in replacement:synonimic hypermutation ratio in CDR regions:

```
ggplot(df) +
  geom_bar(aes(x=pos.aa, weight=weight, fill=region)) +
  geom_density(aes(x=pos.aa, weight=weight, linetype = mutation.type)) +
  ylab("Hypermutation density") +
  xlab("Position in IG, AA") +
  scale_fill_brewer(palette = "Set1") +
  theme_bw()
```

```
## Warning in density.default(x, weights = w, bw = bw, adjust = adjust, kernel
## = kernel, : sum(weights) != 1 -- will not get true density
```

```
## Warning in density.default(x, weights = w, bw = bw, adjust = adjust, kernel
## = kernel, : sum(weights) != 1 -- will not get true density
```



Comparative analysis

Summarize by region and type, compute frequencies and R:S ratio

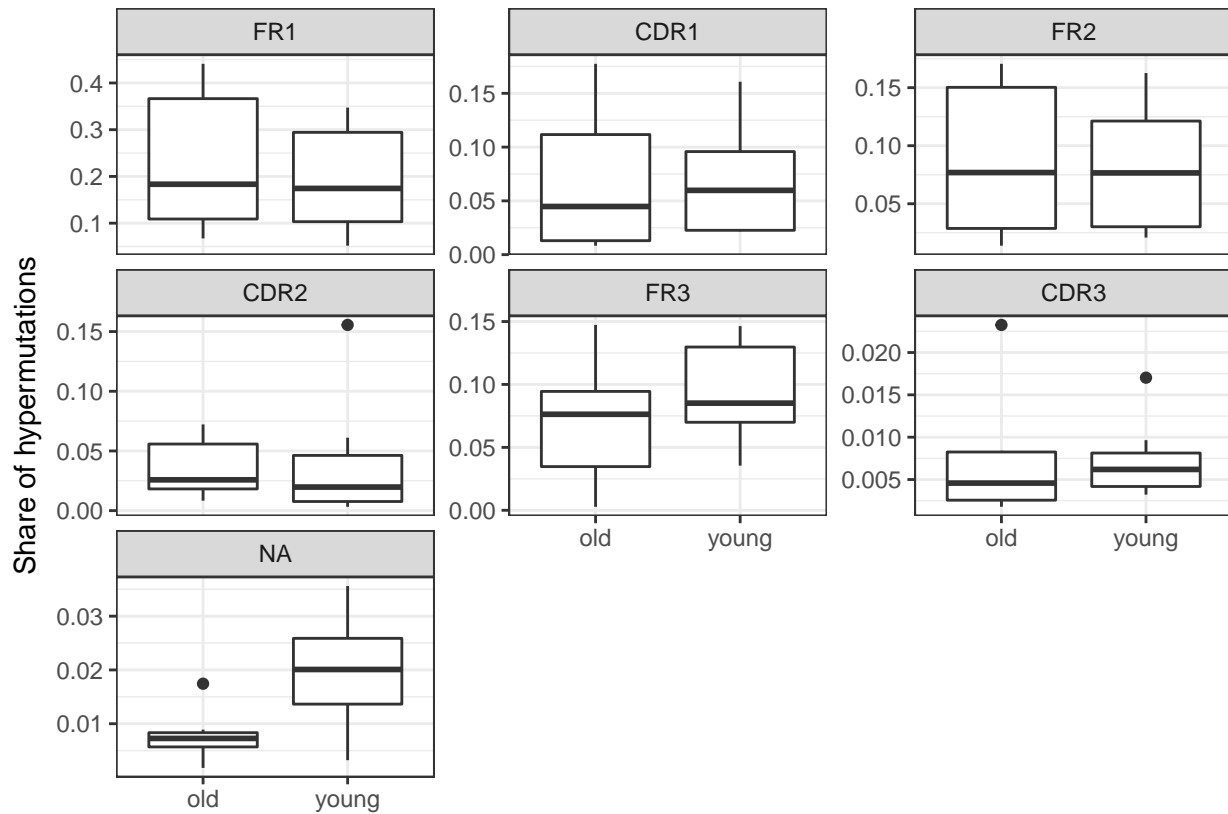
```
df.1 <- ddply(df, .(proj, sample, type, replica, mutation.type, region), summarize,
  count = sum(total.clonotypes))

df.1 <- ddply(df.1, .(proj, sample, type, replica), transform,
  freq = count / sum(count))

df.1 <- ddply(df.1, .(proj, sample, type, replica, region), transform,
  ratio = count / (sum(count) - count))
```

Fraction of errors in each region

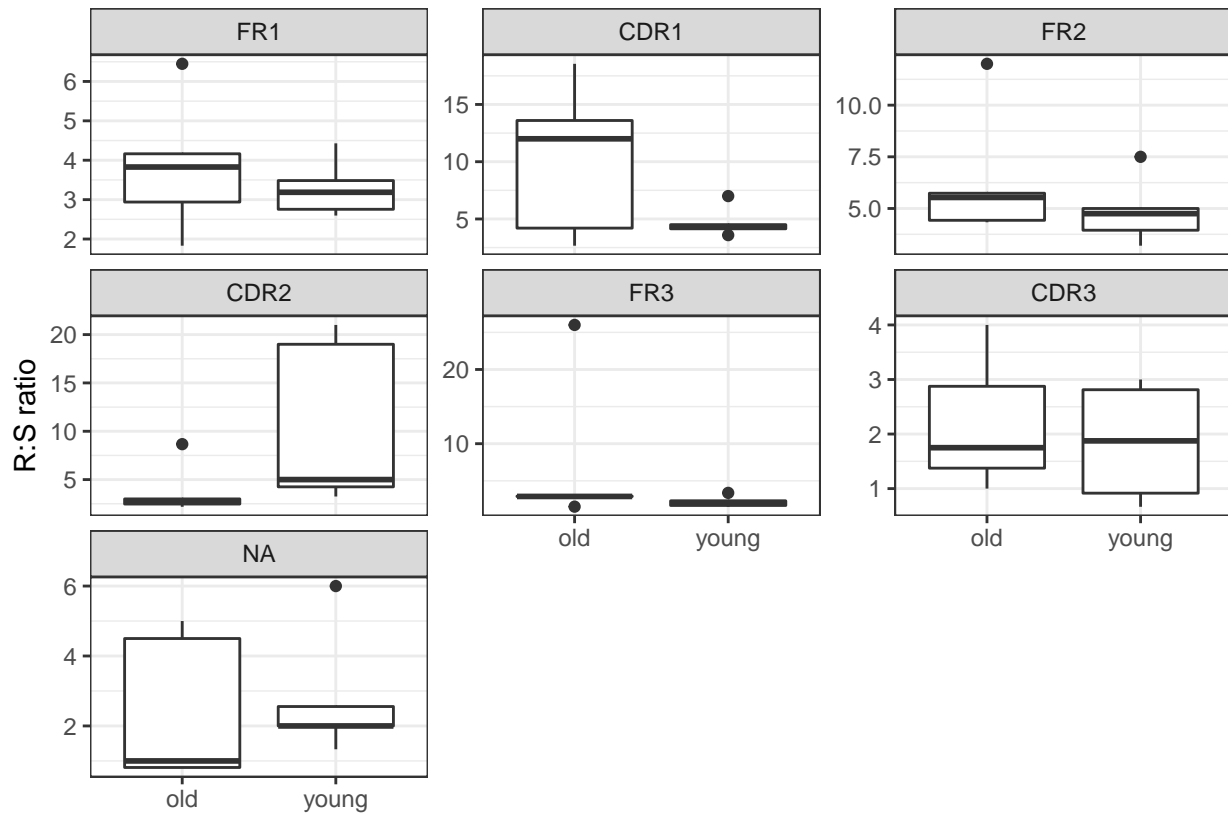
```
ggplot(df.1, aes(x=proj, y = freq)) + geom_boxplot() +
  facet_wrap(~region, scales = "free_y") +
  xlab("") + ylab("Share of hypermutations") +
  theme_bw()
```



R:S ratio varies greatly by region

```
ggplot(subset(df.1, mutation.type == "R"), aes(x=proj, y = ratio)) +
  geom_boxplot() +
  facet_wrap(~region, scales = "free_y") +
  xlab("") + ylab("R:S ratio") +
  theme_bw()
```

Warning: Removed 3 rows containing non-finite values (stat_boxplot).

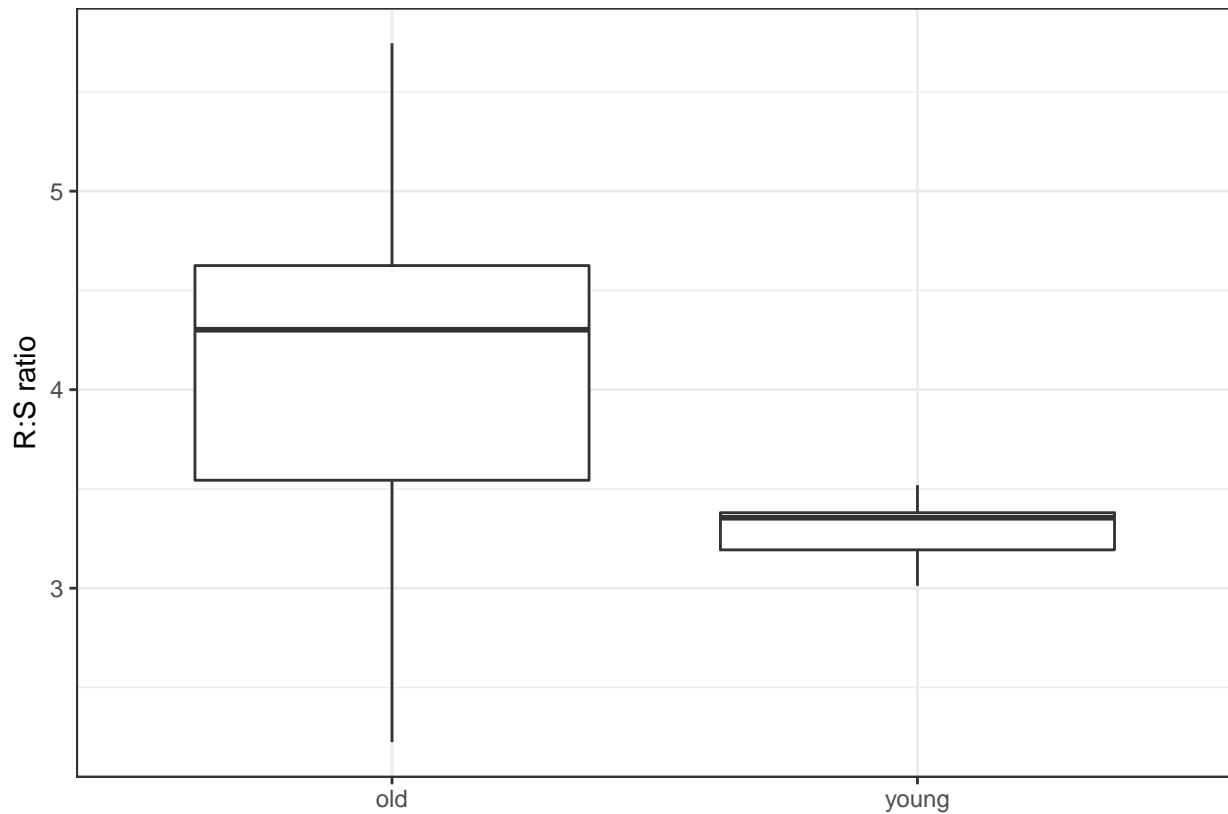


Overall R:S ratio is higher in young

```
df.2 <- ddpoly(df, .(proj, sample, type, replica, mutation.type), summarize,
               count = sum(total.clonotypes))

df.2 <- ddpoly(df.2, .(proj, sample, type, replica), transform,
               ratio = count / (sum(count) - count))

ggplot(subset(df.2, mutation.type == "R"), aes(x=proj, y = ratio)) +
  geom_boxplot() +
  xlab("") + ylab("R:S ratio") +
  theme_bw()
```



```
t.test(ratio ~ proj, subset(df.2, mutation.type == "R"))
```

```
##
## Welch Two Sample t-test
##
## data: ratio by proj
## t = 1.346, df = 4.1765, p-value = 0.2467
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.8191264 2.4116086
## sample estimates:
## mean in group old mean in group young
## 4.088160 3.291919
```

Role of age factor can be deduced using generalized linear model for replacement hypermutation probability (binomial family). Probability of replacement hypermutations is increased by $7 \pm 1\%$ in young compared to old ($P < 10^{-6}$)

```
df$R <- ifelse(df$mutation.type == "R", 1, 0)
fit <- glm(R ~ region + proj - 1, df, family = binomial())
```

```
summary(fit)
```

```
##
## Call:
## glm(formula = R ~ region + proj - 1, family = binomial(), data = df)
##
## Deviance Residuals:
## Min 1Q Median 3Q Max
```

```
## -1.8275    0.6459    0.7181    0.7617    0.8818
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## regionFR1    1.12102    0.07312  15.331 < 2e-16 ***
## regionCDR1    1.46134    0.11871  12.310 < 2e-16 ***
## regionFR2    1.32908    0.10253  12.963 < 2e-16 ***
## regionCDR2    1.22369    0.13973   8.758 < 2e-16 ***
## regionFR3    0.89798    0.08321  10.792 < 2e-16 ***
## regionCDR3    0.77630    0.26650   2.913  0.00358 **
## projyoung   -0.03213    0.07753  -0.414  0.67859
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 5129.3  on 3700  degrees of freedom
## Residual deviance: 4085.4  on 3693  degrees of freedom
## (126 observations deleted due to missingness)
## AIC: 4099.4
##
## Number of Fisher Scoring iterations: 4
```

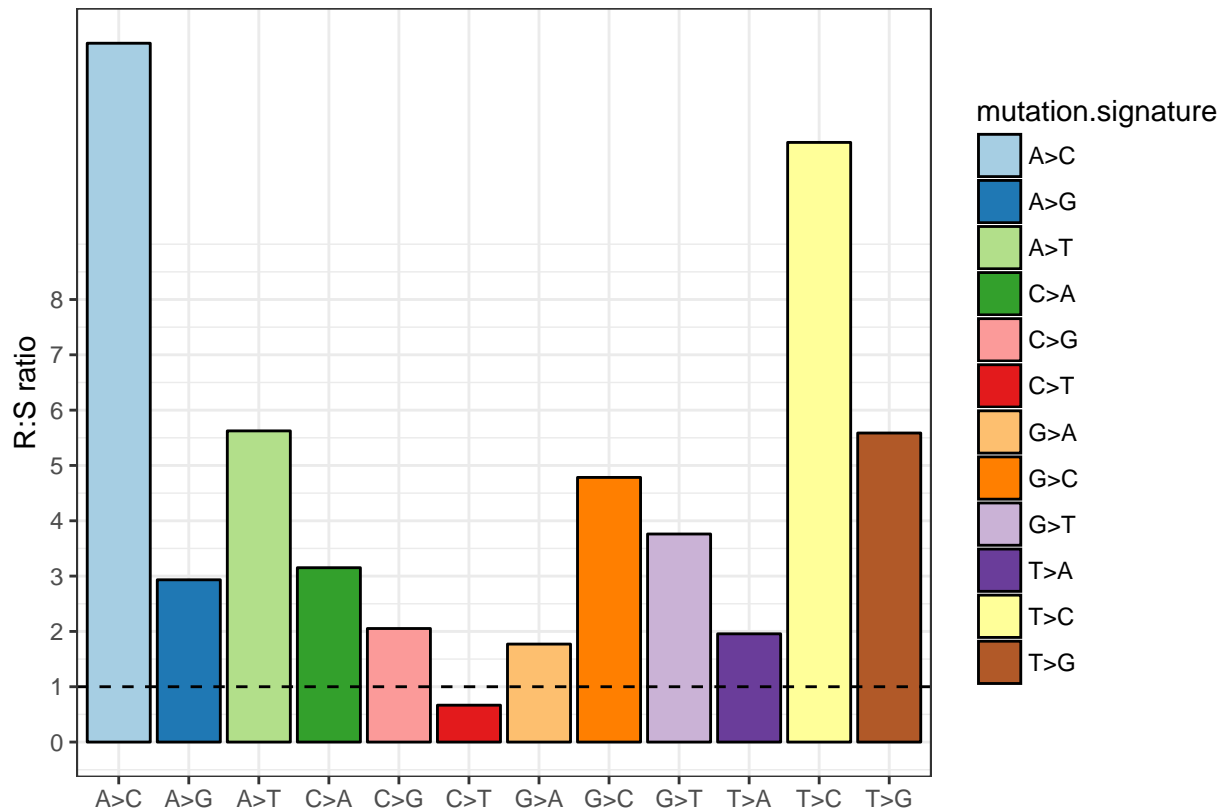
Substitution patterns

R:S ratio across different substitution patterns at nucleotide level

```
df$mutation.signature <- paste(df$from.nt, df$to.nt, sep = ">")

df.3 <- ddply(df, .(mutation.signature, mutation.type), summarize, count = sum(total.clonotypes))
df.3 <- ddply(df.3, .(mutation.signature), transform, ratio = count / (sum(count) - count))
df.3 <- subset(df.3, mutation.type == "R")

ggplot(df.3, aes(x = mutation.signature, y = ratio, fill = mutation.signature)) +
  geom_bar(stat="identity", color="black") +
  geom_hline(yintercept = 1, linetype="dashed") +
  scale_fill_brewer(palette = "Paired") +
  scale_y_continuous("R:S ratio", breaks=0:8) +
  xlab("") +
  theme_bw()
```



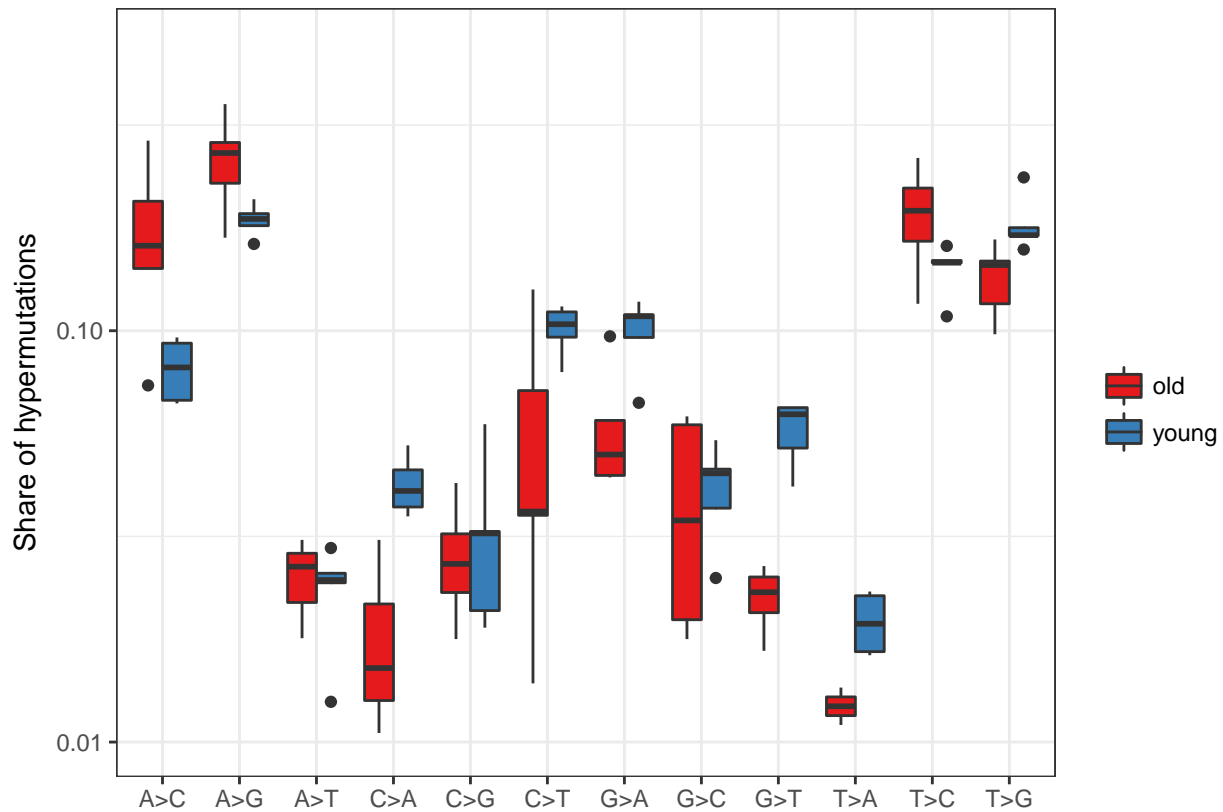
Share of different substitution patterns compared between young and old. Age-related difference observed for certain substitution patterns across hypermutations.

```
df.4 <- ddpoly(df, .(proj, sample, type, replica, mutation.signature), summarize,
  count = sum(total.clonotypes))

df.4 <- ddpoly(df.4, .(proj, sample, type, replica), transform,
  share = count / sum(count))

ggplot(df.4, aes(x = mutation.signature, group = interaction(mutation.signature, proj),
  y = share, fill = proj)) +
  geom_boxplot() + scale_y_log10(limits = c(0.01, 0.5)) +
  scale_fill_brewer("", palette = "Set1") +
  xlab("") + ylab("Share of hypermutations") +
  theme_bw()
```

```
## Warning: Removed 11 rows containing non-finite values (stat_boxplot).
```



```
a <- aov(share ~ mutation.signature * proj, df.4)
summary(a)
```

```
##               Df Sum Sq Mean Sq F value    Pr(>F)
## mutation.signature    11 0.5384  0.04894   53.817 < 2e-16 ***
## proj                    1 0.0000  0.00000    0.000     1
## mutation.signature:proj 11 0.0621  0.00564    6.204 1.26e-07 ***
## Residuals              96 0.0873  0.00091
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hydropathy change patterns observed at amino acid level. More hydrophilic -> hydrophobic amino acid hypermutations in young compared to old.

```
aa.classes <- data.frame(aa = strsplit("I,V,L,F,C,M,A,W,G,T,S,Y,P,H,N,D,Q,E,K,R", ",")[1],
  hydrop = c(rep("hydrophobic", 8), rep("neutral", 6),
    rep("hydrophilic", 6)))

aa.classes$hydrop <- factor(aa.classes$hydrop, c("hydrophobic", "neutral", "hydrophilic"))

df.5 <- ddply(df, .(proj, sample, type, replica, from.aa, to.aa), summarize,
  count = sum(total.clonotypes))

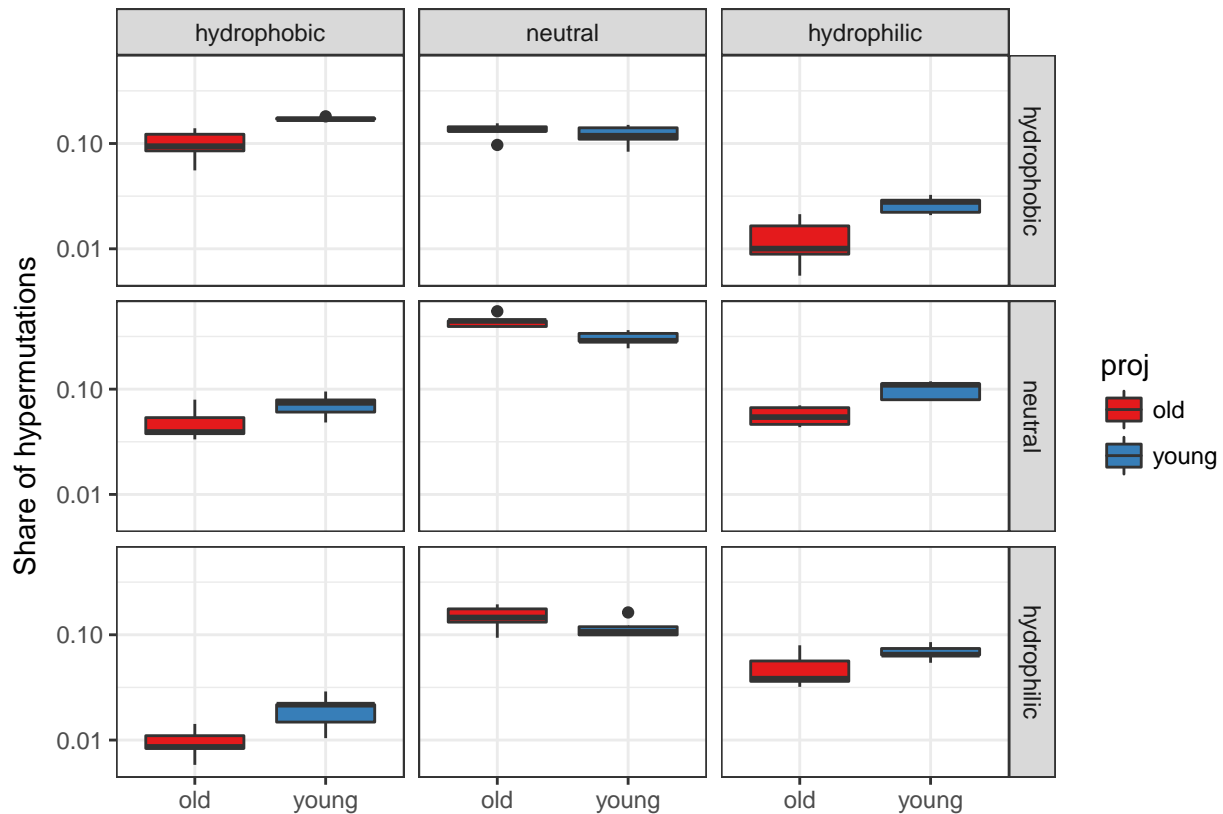
df.5 <- merge(df.5, aa.classes, by.x = "from.aa", by.y = "aa")
df.5 <- merge(df.5, aa.classes, by.x = "to.aa", by.y = "aa")
#df.5$signature <- paste(df.5$hydrop.x, df.5$hydrop.y, sep = ">")

df.5 <- ddply(df.5, .(proj, sample, type, replica, hydrop.x, hydrop.y), summarize,
  count = sum(count))
```



```
df.5 <- ddply(df.5, .(proj, sample, type, replica), transform,
              share = count / sum(count))

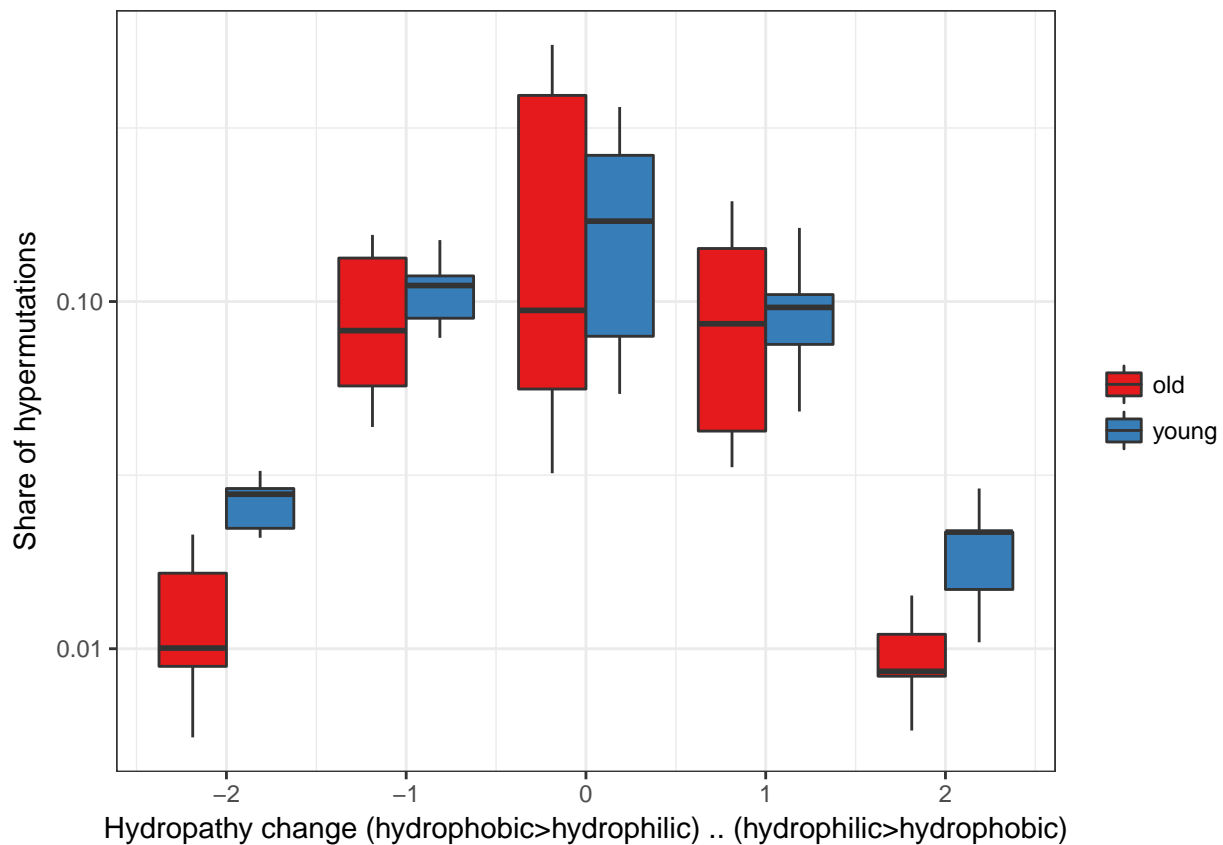
ggplot(df.5, aes(x=proj, group=proj, y=share, fill = proj)) + geom_boxplot() +
  facet_grid(hydrop.x~hydrop.y) + scale_y_log10() +
  scale_fill_brewer(palette = "Set1") +
  xlab("") + ylab("Share of hypermutations") +
  theme_bw()
```



```
hydrop_toint <- function(x) {
  ifelse(x == "hydrophobic", 1, ifelse(x == "neutral", 0, -1))
}

df.5$hydrop.change <- with(df.5, hydrop_toint(hydrop.y) - hydrop_toint(hydrop.x))

ggplot(df.5, aes(x=hydrop.change, group = interaction(hydrop.change, proj), y=share,
              fill=proj)) +
  geom_boxplot() + scale_y_log10() +
  xlab("Hydropathy change (hydrophobic>hydrophilic) .. (hydrophilic>hydrophobic)") +
  ylab("Share of hypermutations") +
  scale_fill_brewer("", palette = "Set1") +
  theme_bw()
```



```
a <- aov(share ~ hydrop.x : hydrop.y + hydrop.x : hydrop.y : proj, df.5)
summary(a)
```

```
##               Df Sum Sq Mean Sq F value    Pr(>F)
## hydrop.x:hydrop.y      8 0.9471  0.11838  168.34 < 2e-16 ***
## hydrop.x:hydrop.y:proj  9 0.0737  0.00818   11.64 4.64e-11 ***
## Residuals              72 0.0506  0.00070
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#not enough 'x' observations
#t.test(share ~ proj,
#       subset(df.5, hydrop.y == "hydrophobic" & hydrop.x == "hydrophilic"))
```