Polymerase fidelity estimates

Load required libraries, load and merge MAGERI results.

```
library(plyr); library(ggplot2); library(reshape2); library(gplots)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.4
```

```
## Warning: package 'gplots' was built under R version 3.2.4
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
df.meta <- read.table("metadata.txt", header=T, sep = "\t")

df <- data.frame()

for (project in levels(df.meta$project)) {
  for (sample in levels(df.meta$sample)) {
    fname <- paste(project, sample, "variant.caller.txt", sep =".")
    if (file.exists(fname)) {
      df.1 <- read.table(fname, header=T, sep="\t")
      df.1$project <- project
      df.1$sample <- sample
      df <- rbind(df, df.1)
    }
  }
}

df$project <- as.factor(df$project)
df$sample <- as.factor(df$sample)


df <- merge(df, df.meta, all.x=T, all.y=F)

template <- "TGGGATCCATTATCGGCGGCGAATTTACCACCATTGAAAACCAGCCGTGGTTTGCGGCGATTTATCGTCGTCATCGTGGCGGCAGCGTGA
```

Error rates

Overall error rates

```
library(knitr)

df.er <- ddply(df, .(project, name), summarize,
               mismatches = sum(count.major), umi.count = round(mean(coverage)),
```

```
                err.rate = mismatches / umi.count / nchar(template) / mean(cycles),
                delta = sqrt(mismatches / umi.count * (1 - mismatches / umi.count) / umi.count) / nchar(t
                err.lb = err.rate - 1.96 * delta, err.ub = err.rate + 1.96 * delta)

kable(df.er)
```

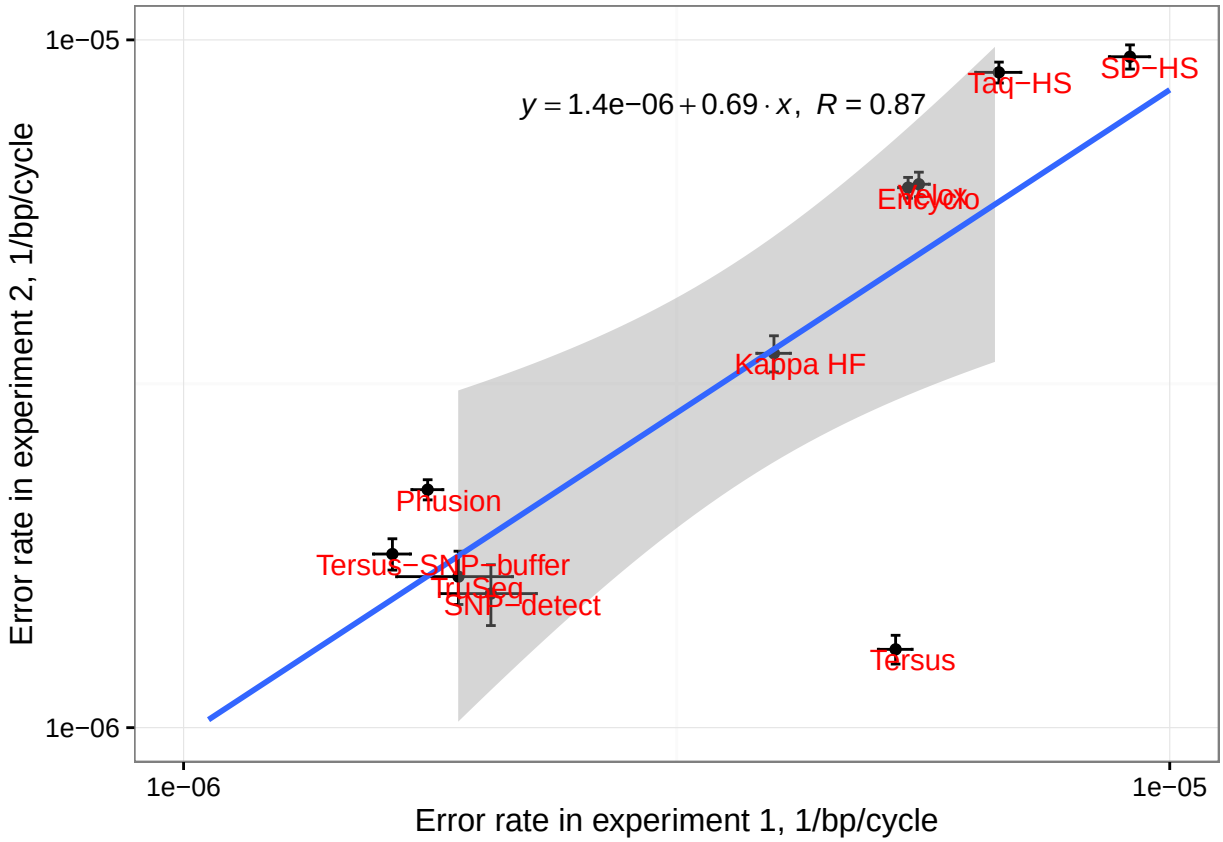
project	name	mismatches	umi.count	err.rate	delta	err.lb	err.ub
polerr73	Encyclo	7759	558336	5.4e-06	1e-07	5.3e-06	5.5e-06
polerr73	Kappa HF	2444	240548	4.0e-06	1e-07	3.8e-06	4.1e-06
polerr73	Phusion	2902	512908	1.8e-06	0e+00	1.7e-06	1.8e-06
polerr73	SD-HS	1696	72692	9.1e-06	2e-07	8.7e-06	9.5e-06
polerr73	SNP-detect	303	57755	2.0e-06	1e-07	1.8e-06	2.3e-06
polerr73	Taq-HS	1351	78646	6.7e-06	2e-07	6.4e-06	7.1e-06
polerr73	Tersus	2430	180029	5.3e-06	1e-07	5.1e-06	5.5e-06
polerr73	Tersus-SNP-buffer	2035	488168	1.6e-06	0e+00	1.6e-06	1.7e-06
polerr73	TruSeq	211	43380	1.9e-06	1e-07	1.6e-06	2.2e-06
polerr73	Velox	6089	427259	5.6e-06	1e-07	5.4e-06	5.7e-06
polerr82	Encyclo	3127	200300	6.1e-06	1e-07	5.9e-06	6.3e-06
polerr82	Kappa HF	1040	116021	3.5e-06	1e-07	3.3e-06	3.7e-06
polerr82	Phusion	3349	471735	2.2e-06	0e+00	2.1e-06	2.3e-06
polerr82	SD-HS	2294	94810	9.5e-06	2e-07	9.1e-06	9.8e-06
polerr82	SNP-detect	372	92765	1.6e-06	1e-07	1.4e-06	1.7e-06
polerr82	Taq-HS	3076	133952	9.0e-06	2e-07	8.7e-06	9.3e-06
polerr82	Tersus	1655	497406	1.3e-06	0e+00	1.2e-06	1.4e-06
polerr82	Tersus-SNP-buffer	1391	303834	1.8e-06	0e+00	1.7e-06	1.9e-06
polerr82	TruSeq	483	113803	1.7e-06	1e-07	1.5e-06	1.8e-06
polerr82	Velox	2234	141485	6.2e-06	1e-07	5.9e-06	6.4e-06

Error rate consistency

```
df.er.cast <- dcast(df.er, name ~ project,
                    value.var = "err.rate")
df.er.cast2 <- dcast(df.er, name ~ project,
                    value.var = "delta")

df.er.cast <- merge(df.er.cast, df.er.cast2, by = "name")

m <- lm(polerr73.x ~ polerr82.x, df.er.cast);
eq <- substitute(italic(y) == a + b %.% italic(x)*","~~italic(R)~"="~r,
     list(a = format(coef(m)[1], digits = 2),
          b = format(coef(m)[2], digits = 2),
          r = format(sqrt(summary(m)$r.squared), digits = 2)))
lbl<-as.character(as.expression(eq))

ggplot(df.er.cast, aes(x=polerr73.x, y=polerr82.x)) + geom_point() +
  geom_errorbarh(aes(xmax=polerr73.x+1.96*polerr73.y,xmin=polerr73.x-1.96*polerr73.y)) +
  geom_errorbar(aes(ymax=polerr82.x+1.96*polerr82.y,ymin=polerr82.x-1.96*polerr82.y)) +
  geom_smooth(method = "lm", fullrange = T) +
  geom_text(aes(label=name), vjust=1, hjust = .3, color="red") +
  annotate("text", x = 2e-6, y = 8e-6, label = lbl, hjust=-0.1, parse = TRUE)+
  scale_x_log10(name="Error rate in experiment 1, 1/bp/cycle", limits=c(1e-6,1e-5)) +
```

```
  scale_y_log10(name="Error rate in experiment 2, 1/bp/cycle", limits=c(1e-6,1e-5)) +
  theme_bw()
```

```
## Warning: Removed 2 rows containing missing values (geom_smooth).
```



Combine error rates from a pair of experiments.

```
df.coverage.summary <- ddply(df, .(name, project), summarize, coverage = mean(coverage))
df.coverage.summary <- ddply(df.coverage.summary, .(name), summarize, coverage = sum(coverage))

df <- ddply(df, .(name, mutation), summarize, count.major = sum(count.major))

df <- merge(df, df.coverage.summary, by = "name")
```

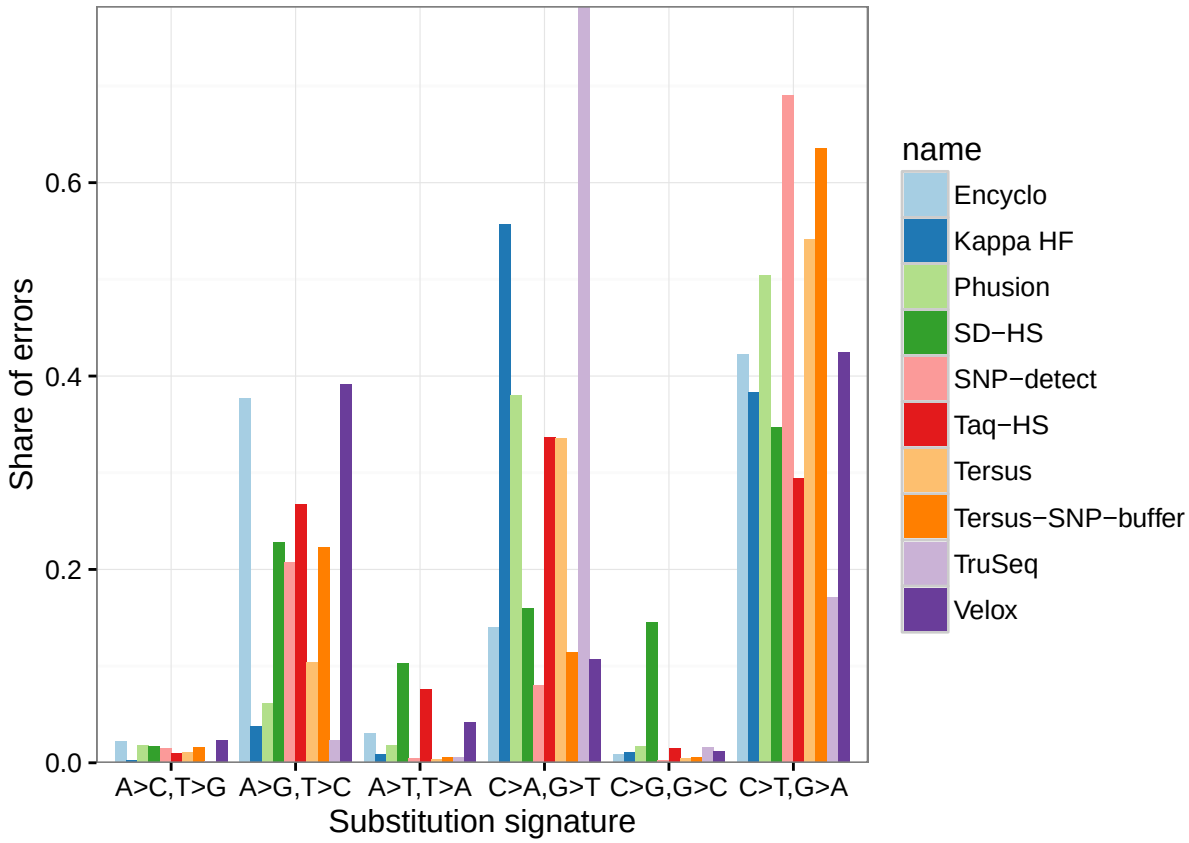
Error substitution patterns

Parse mutation signatures (needed for further analysis)

```
df$mut.split <- sapply(df$mutation, function(x) strsplit(as.character(x),"[S:>]"))
df$mutation.pos <- as.integer(sapply(df$mut.split, function(x) x[2]))
df$mutation.from <- sapply(df$mut.split, function(x) x[3])
df$mutation.to <- sapply(df$mut.split, function(x) x[4])
df$mut.split  <- NULL
```

Substitution signature preferences

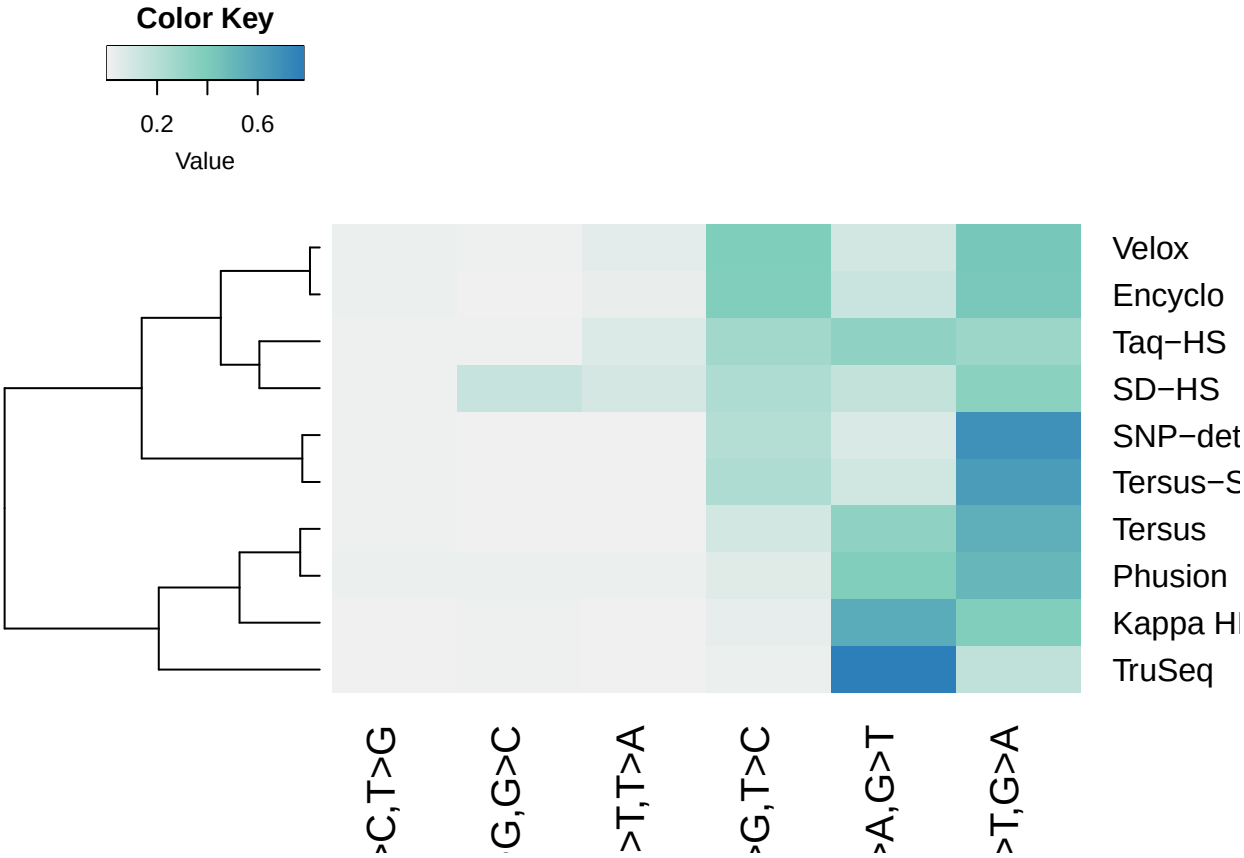
```
df$mutation.signature <- paste(df$mutation.from, df$mutation.to, sep =">")

sign.rep <- data.frame(mutation.signature = c("A>C","A>G","A>T","C>A","C>G","C>T","G>A","G>C","G>T","T>
                       mutation.signature.rep = c("A>C,T>G","A>G,T>C","A>T,T>A","C>A,G>T","C>G,G>C","C>

df <- merge(df, sign.rep, all.x=T, all.y=F)

df.pattern <- ddply(df, .(name, mutation.signature.rep), summarize, count.sum = sum(count.major))
df.pattern <- ddply(df.pattern, .(name), transform, freq = count.sum / sum(count.sum))

ggplot(df.pattern, aes(x = mutation.signature.rep, weight = freq,
                fill = name)) + geom_bar(position = position_dodge()) +
  xlab("Substitution signature") + scale_y_continuous("Share of errors", expand=c(0,0)) +
  scale_fill_brewer(palette = "Paired") + theme_bw()
```




```
df.pattern.mat <- dcast(df.pattern, name ~ mutation.signature.rep, value.var = "freq")
rownames(df.pattern.mat) <- df.pattern.mat$name
df.pattern.mat$name <- NULL

df.pattern.mat <- as.matrix(df.pattern.mat)

heatmap.2(df.pattern.mat, col=colorpanel(100, "#f0f0f0", "#7fcdbb", "#2c7fb8"),
          hclustfun = function(d) hclust(d, method="ward"),
```

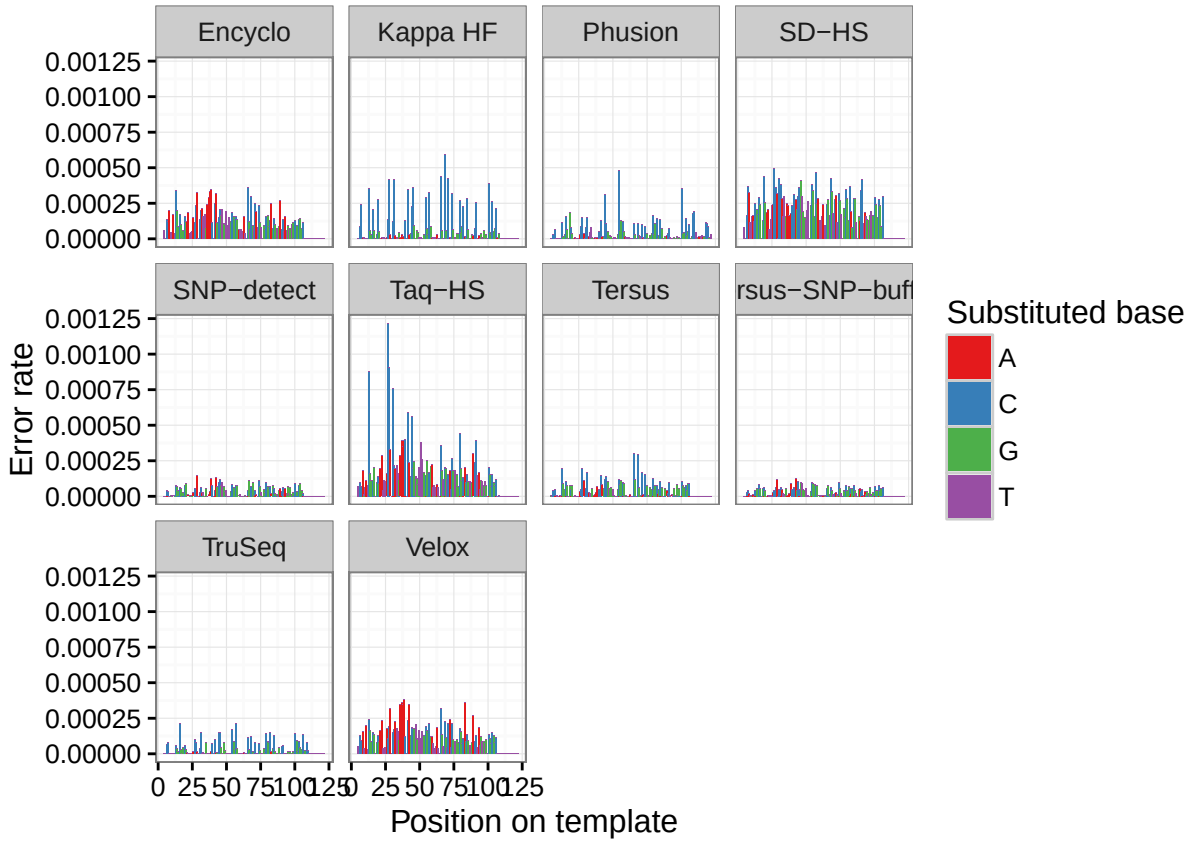
```
          dendrogram = "row",
          density.info = "none", trace="none")
```

```
## 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"
```



This goes to supplementary. Clear preference of substitution patters for different polymerases, but no position-related trend.

```
ggplot(df, aes(x = mutation.pos, weight = count.major / coverage, fill = mutation.from)) +
  geom_histogram(bins = nchar(template)) + scale_fill_brewer("Substituted base", palette = "Set1") +
  xlab("Position on template") + ylab("Error rate") +
  facet_wrap(~name) + theme_bw()#, scales = "free_y") + theme_bw()
```




```
a <- aov(count.major / coverage ~ mutation.from * name + mutation.pos, df)
summary(a)
```

```
##                       Df    Sum Sq   Mean Sq F value   Pr(>F)
## mutation.from          3 5.840e-07 1.946e-07  44.781  < 2e-16 ***
## name                   9 1.064e-06 1.182e-07  27.207  < 2e-16 ***
## mutation.pos           1 1.000e-08 1.036e-08   2.384    0.123
## mutation.from:name    27 5.010e-07 1.857e-08   4.273 1.72e-12 ***
## Residuals           2121 9.217e-06 4.350e-09
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

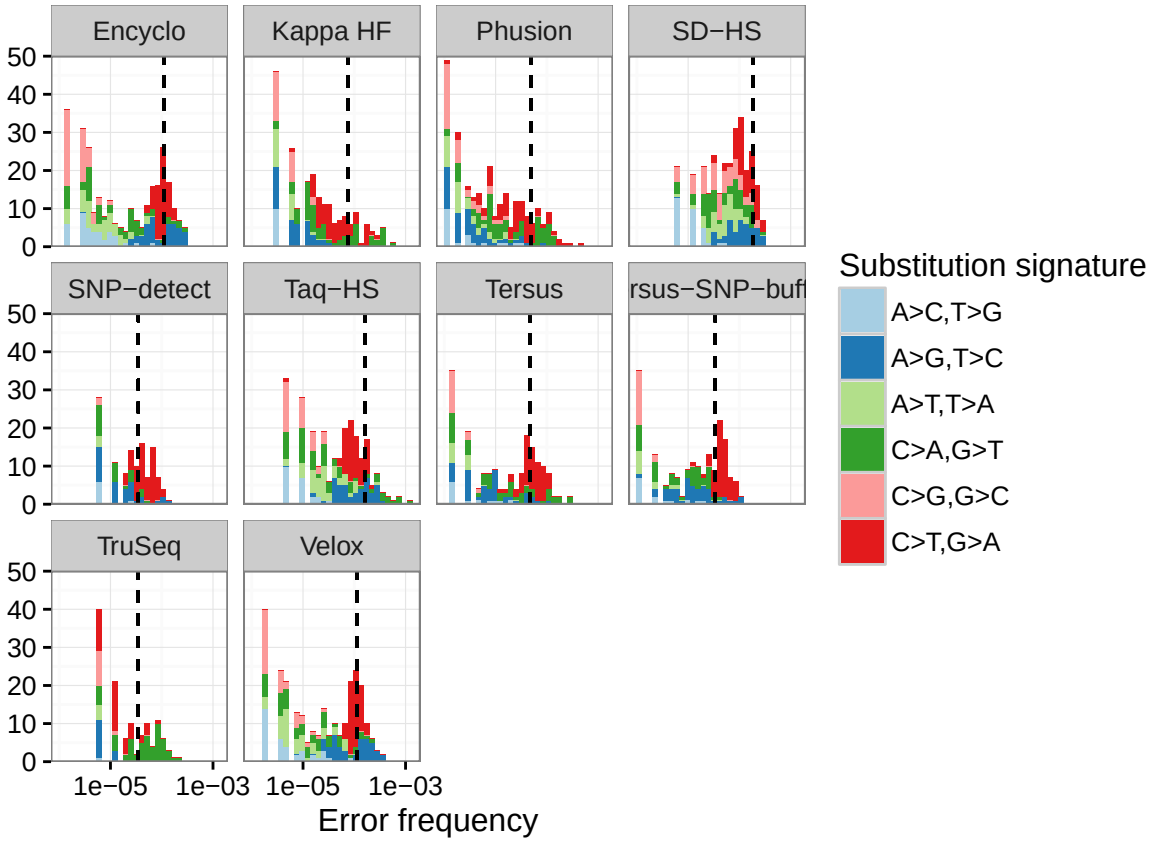
Error hotspot context pattern

Frequency distribution of individual mutations, grouped by their pattern.

```
df.mean.err <- ddply(df, .(name), summarize, mean.err.rate = sum(count.major) / mean(coverage) / nchar(

df <- merge(df, df.mean.err, all.x=T, all.y=F)

ggplot(df) +
  geom_histogram(aes(x = count.major / coverage, fill = mutation.signature.rep)) +
  geom_linerange(aes(x = mean.err.rate, ymin = 0, ymax=50), linetype = "dashed", color="black") +
  scale_fill_brewer("Substitution signature", palette = "Paired") +
```

```
  scale_x_log10("Error frequency") +
  scale_y_continuous("", expand=c(0,0)) + facet_wrap(~name) + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

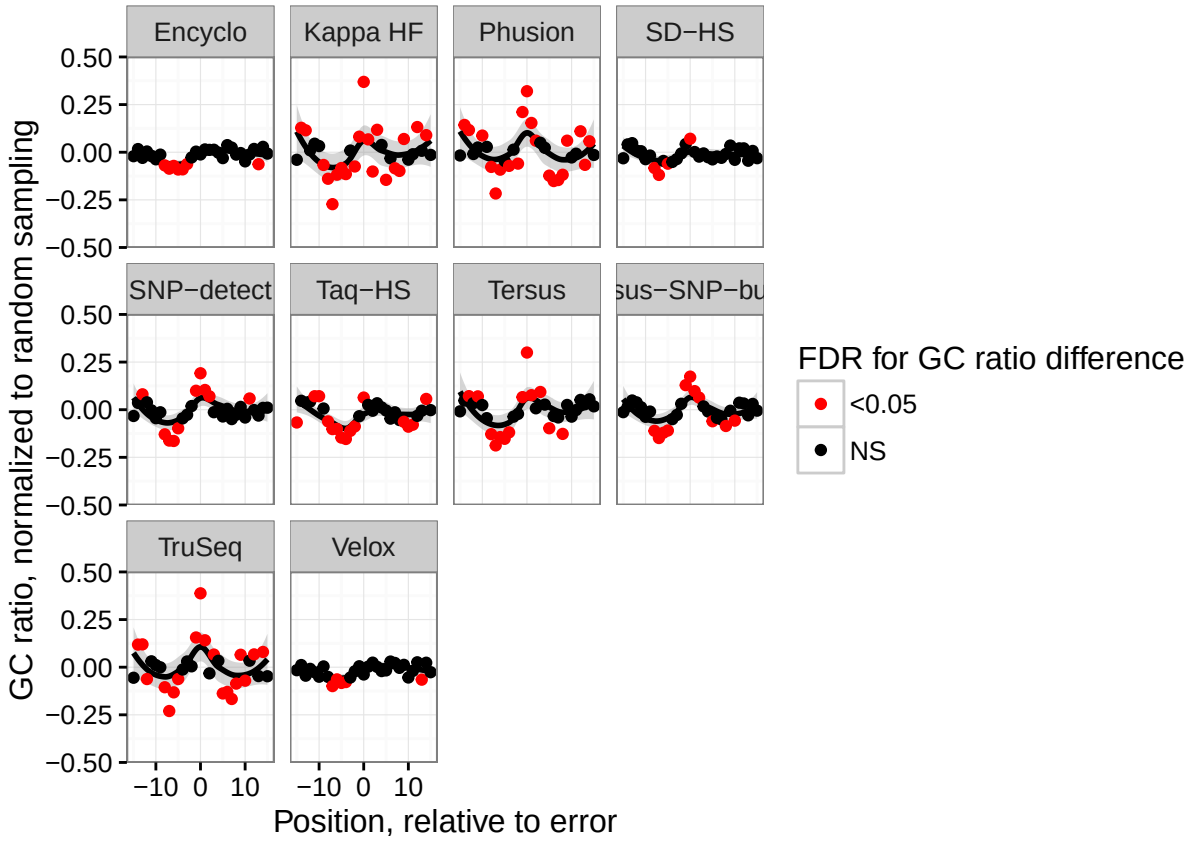



```
  #facet_wrap(~name, scales = "free_y") + theme_bw()
```

GC context profile.

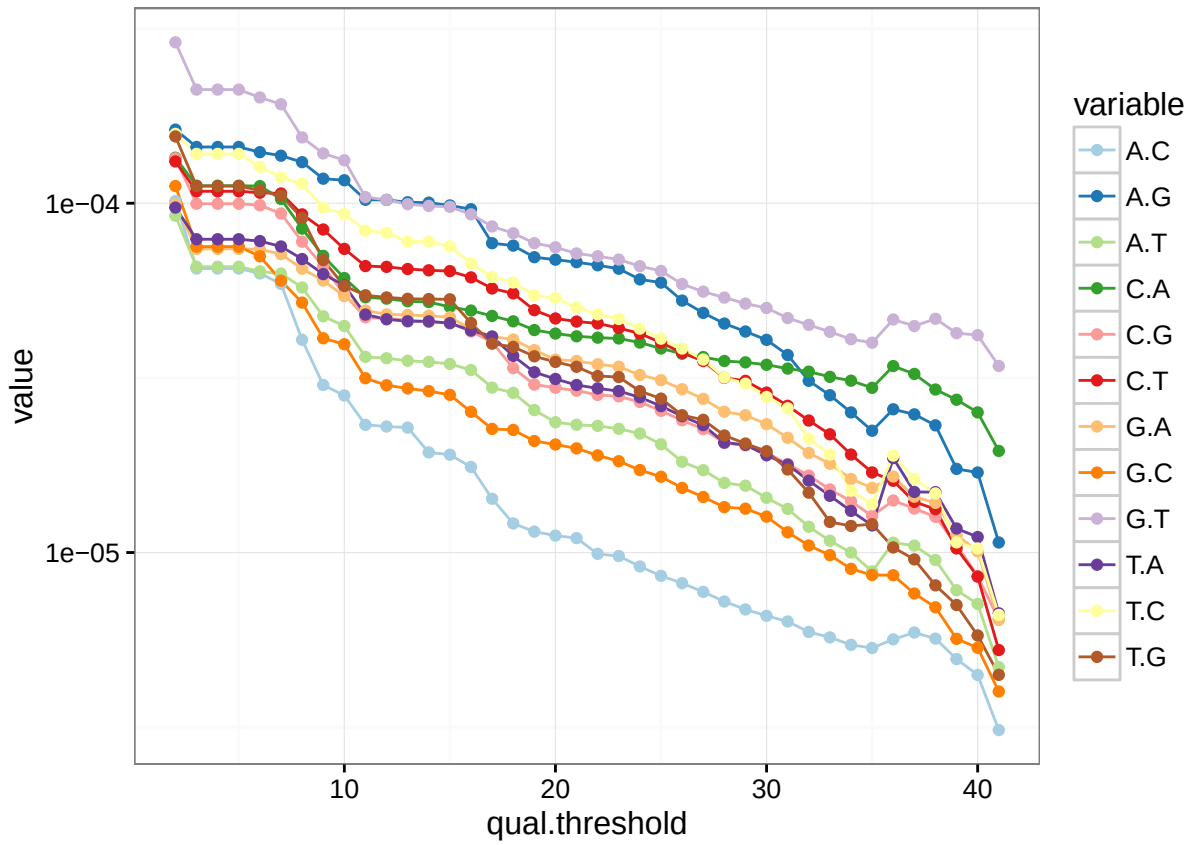
```
window.size <- 15
padding <- paste(rep("N", window.size), sep ="", collapse="")
template.p <- paste(padding, template, padding, sep="")

df$context <- sapply(df$mutation.pos,
                     function(pos) substring(template.p, pos + 1, pos + 2 * window.size + 1))

df.context <- ddply(df, .(name, context), summarize,
                    weight = sum(count.major))

context.rand <- sapply(sample(1:nchar(template), 1000, replace=T),
                     function(pos) substring(template.p, pos + 1, pos + 2 * window.size + 1))

df.context.rand <- data.frame(context = context.rand)
df.context.rand$name <- "random"
```

```
df.context.rand$weight <- 1

df.context <- rbind(df.context, df.context.rand)

write.table(df.context, file = "context.txt", quote = F, sep = "\t", row.names = F)

system("groovy ProcessContext.groovy")

df.context.profile <- read.table("context.proc.txt", header=T, sep="\t")

df.context.normalized <- merge(subset(df.context.profile, name != "random"),
                               subset(df.context.profile, name == "random"),
                               by = "pos")

df.context.normalized <- ddply(df.context.normalized, .(pos, name.x), transform,
                               pval = prop.test(x = value.y, n = sum.y, p = value.x / sum.x,
                                                alternative = "two.sided", correct = F)[[3]])

df.context.normalized$pval <- p.adjust(df.context.normalized$pval)

ggplot(df.context.normalized, aes(x = pos - window.size,
                                  y = value.x / sum.x - value.y / sum.y)) +
  geom_smooth(colour="black") + geom_point(aes(color = factor(ifelse(pval < 0.05, "<0.05", "NS")))) +
  facet_wrap(~name.x) + scale_colour_manual("FDR for GC ratio difference", values=c("red", "black")) +
  scale_y_continuous("GC ratio, normalized to random sampling", limits = c(-0.5, 0.5), expand=c(0,0)) +
  theme_bw()
```



Sequence quality and error patterns

```
df.q <- read.table("mmqc.txt", header=T,sep="\t")
library(reshape2)
df.q <- melt(df.q, id.vars = "qual.threshold")

ggplot(df.q, aes(x=qual.threshold, color=variable, y=value)) +
  geom_line() + geom_point() + scale_y_log10() +
  scale_color_brewer(palette = "Paired") + theme_bw()
```



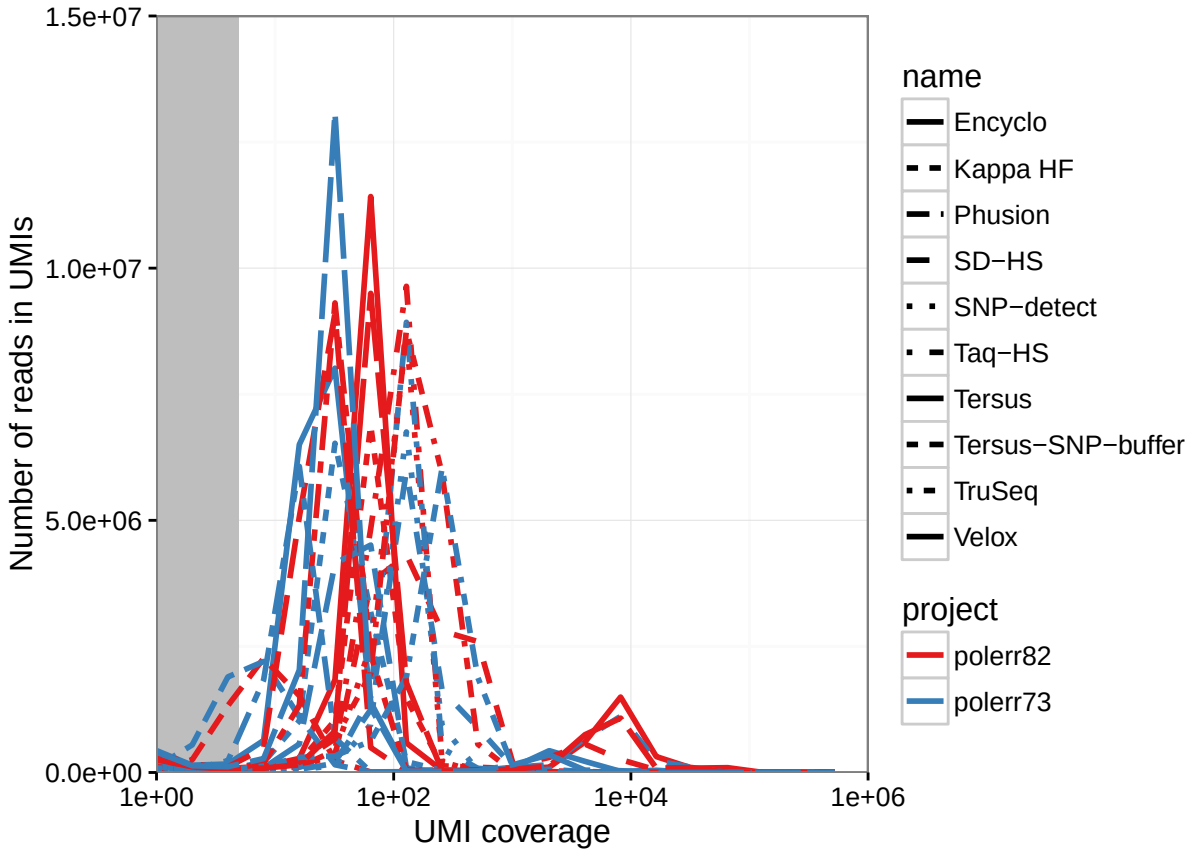

```
df.q$subset <- ifelse(df.q$qual.threshold >= 35, "high-quality", "low-quality")
df.q$signature <- ifelse(df.q$variable %in% c("G.T", "C.A"), "TrueSeq", "other")

#ggplot(df.q, aes(x=subset, group=interaction(subset, signature), fill=signature, y = value)) +
#  geom_boxplot()
```

UMI coverage

UMI coverage histogram

```
df.h <- data.frame(mig.size.bin = integer(), read.count = integer(), name=character(), project=characte

for (proj in unique(df.meta$project)) {
  for (sample in unique(df.meta$sample)) {
    mask <- which(df.meta$sample == sample & df.meta$project == proj)
    name <- df.meta$name[mask][1]
    cycles_2 <- df.meta$cycles_2[mask][1]
    dna_2 <- df.meta$dna_2[mask][1]
    df.hh <- read.table(paste(proj, sample, "umi.histogram.txt", sep="."), header=T, sep="\t")
    df.h <- rbind(df.h, data.frame(mig.size.bin = df.hh$mig.size.bin, read.count = df.hh$read.count,
                                   name=name, project=proj, cycles_2=cycles_2, dna_2=dna_2))
  }
}
```

```
ggplot(df.h) +
  geom_rect(aes(xmin=1, xmax=5, ymin=0, ymax=Inf), fill="grey") +
  geom_line(aes(x=mig.size.bin, y=read.count, color=project, linetype=name,
                group = interaction(project, name)), size=1) +
  scale_x_log10("UMI coverage", limits = c(1,1e6), expand=c(0,0)) +
  scale_y_continuous("Number of reads in UMIs", expand=c(0,0), limits=c(0,1.5e7)) +
  scale_color_brewer(palette = "Set1") +
  theme_bw()
```



Coverage and PCR cycles

```
df.o <- ddply(df.h, .(project, name, dna_2, cycles_2), summarize,
              peak = log2(mig.size.bin[which(read.count == max(read.count))]))

df.o$dna_2<-factor(df.o$dna_2)

m <- lm(peak ~ cycles_2, subset(df.o, dna_2=="0.02"))
eq <- substitute(italic(R)~"="~r*","~~italic(P)~"="~p,
     list(
     r = format(sqrt(summary(m)$r.squared), digits = 2),
     p = format(summary(m)$coefficient[[8]], digits = 3)
     ))
lbl<-as.character(as.expression(eq))

ggplot()+
```

```
geom_label(aes(x = 10, y = 8, label = lbl), hjust=-0.1, parse = TRUE)+
stat_smooth(data=subset(df.o, dna_2=="0.02"), aes(cycles_2, peak), method=lm, color="black", fill="grey
geom_jitter(data=df.o,aes(cycles_2,peak,shape=dna_2,color=name), size=2, width=0.3, height=0.3) +
#geom_text(aes(cycles_2,peak,label=name, vjust=1, hjust = .3)) +
scale_x_continuous(name="2nd PCR cycles",limits=c(10,20), breaks=10:18) +
scale_y_continuous(expression('log'[2]~'characteristic MIG size'), breaks=2:10) +
scale_colour_brewer(name  ="Polymerase", palette = "Paired") +
scale_shape(name  ="DNA amount, ng") +
theme_bw()
```