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
```

```
read_data <- function(path) {
  df.meta <- read.table(paste(path, "metadata.txt", sep="/"), header=T, sep = "\t")

  df <- data.frame()

  for (project in unique(df.meta$project)) {
    for (sample in unique(df.meta$sample)) {
      fname <- paste(path, paste(project, sample, "variant.caller.txt", sep ="."), 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 <- merge(df, df.meta, all.x=T, all.y=F)
}

df <- read_data("polerr2016")
df.1 <- df
df.linpcr <- read_data("polerr73-82")

template <- "TGGGATCCATTATCGGCGGCGAATTTACCACCATTGAAAACCAGCCGTGGTTTGCGGCGATTTATCGTCGTCATCGTGGCGGCAGCGTGA
```

Estimating the error rates

In this section we compute linear PCR error rate from Proj73/82 and error rate in conventional PCR from Polerr2016 project. Note that the linear PCR error rate is substantially higher than rate per cycle of conentional PCR. For some cases, e.g. Phusion it accounts for ~1/2 of errors observed in Polerr2016.

```
library(knitr)

# Compute linear PCR error rate

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

df.er.linpcr <- data.frame(name = df.er.linpcr$name, linpcr.er = df.er.linpcr$mismatches/df.er.linpcr$um

# Compute uncrorrected error rate

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

df.er <- merge(df.er, df.er.linpcr, by = "name", all.x = T)

df.er$err.rate <- with(df.er, mismatches / umi.count / nchar(template) / mean(cycles))
df.er$delta <- with(df.er, sqrt(mismatches / umi.count * (1 - mismatches / umi.count) / umi.count) /
                 nchar(template) / mean(cycles))
df.er$err.lb <- df.er$err.rate - 1.96 * df.er$delta
df.er$err.ub <- df.er$err.rate + 1.96 * df.er$delta

# Error rates corrected for linear PCR errors

df.er$mismatches.corr <- with(df.er, mismatches - linpcr.er * umi.count * nchar(template))
df.er$err.rate.corr <- with(df.er, mismatches.corr / umi.count / nchar(template) / mean(cycles))
df.er$delta.corr <- with(df.er, sqrt(mismatches.corr / umi.count * (1 - mismatches.corr / umi.count) / u
                 nchar(template) / mean(cycles))
df.er$err.lb.corr <- df.er$err.rate.corr - 1.96 * df.er$delta.corr
df.er$err.ub.corr <- df.er$err.rate.corr + 1.96 * df.er$delta.corr

df.er$cycles <- NULL

kable(df.er)
```

name	project	mismatches	umi.count	linpcr.er	err.rate	delta	err.lb	err.ub	misma
Encyclo	polerr2016-1	20184	189566	0.0001121	4.06e-05	3.0e-07	4.00e-05	4.11e-05	17
Encyclo	polerr2016-2	11721	103838	0.0001121	4.30e-05	4.0e-07	4.23e-05	4.38e-05	10
Kappa HF	polerr2016-1	321	8444	0.0000763	1.45e-05	8.0e-07	1.29e-05	1.60e-05	
Kappa HF	polerr2016-2	2402	63892	0.0000763	1.43e-05	3.0e-07	1.38e-05	1.49e-05	1
Phusion	polerr2016-2	22	1388	0.0000496	6.00e-06	1.3e-06	3.50e-06	8.50e-06	
Phusion	polerr2016-1	25	1381	0.0000496	6.90e-06	1.4e-06	4.20e-06	9.60e-06	
SD-HS	polerr2016-2	8450	59856	0.0001861	5.38e-05	5.0e-07	5.27e-05	5.49e-05	7
SD-HS	polerr2016-1	5465	33817	0.0001861	6.16e-05	8.0e-07	6.01e-05	6.31e-05	4
SNP-detect	polerr2016-1	358	14249	0.0000350	9.60e-06	5.0e-07	8.60e-06	1.06e-05	
SNP-detect	polerr2016-2	670	32867	0.0000350	7.80e-06	3.0e-07	7.20e-06	8.40e-06	
Taq-HS	polerr2016-1	1526	15510	0.0001627	3.75e-05	9.0e-07	3.57e-05	3.93e-05	1
Taq-HS	polerr2016-2	2575	24587	0.0001627	3.99e-05	7.0e-07	3.85e-05	4.14e-05	2
Tersus	polerr2016-2	1022	31391	0.0000471	1.24e-05	4.0e-07	1.17e-05	1.32e-05	
Tersus	polerr2016-1	2226	47794	0.0000471	1.77e-05	4.0e-07	1.70e-05	1.85e-05	1
Tersus-SNP-buffer	polerr2016-2	4805	173172	0.0000338	1.06e-05	2.0e-07	1.03e-05	1.09e-05	4
Tersus-SNP-buffer	polerr2016-1	4891	133148	0.0000338	1.40e-05	2.0e-07	1.36e-05	1.44e-05	4
TruSeq	polerr2016-1	605	24258	0.0000345	9.50e-06	4.0e-07	8.80e-06	1.03e-05	

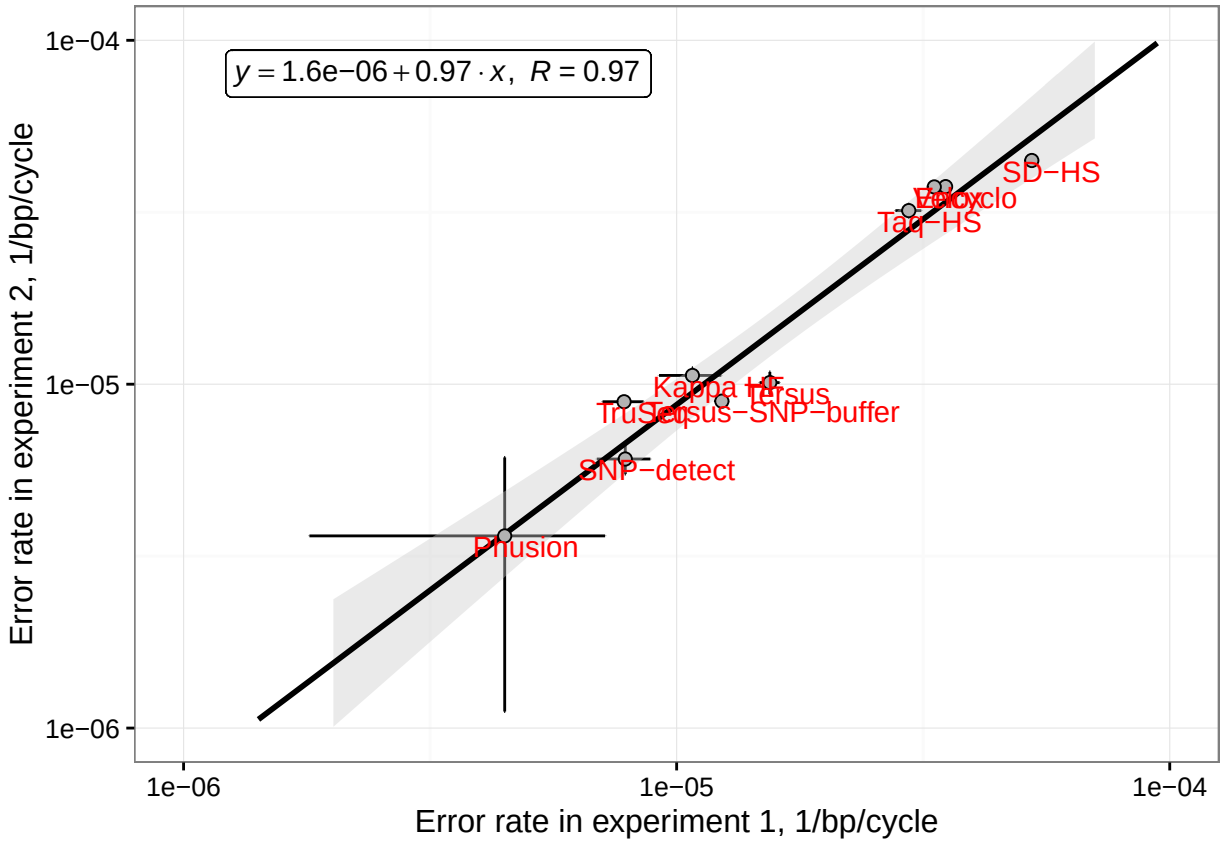
name	project	mismatches	umi.count	linpcr.er	err.rate	delta	err.lb	err.ub	misma
TruSeq	polerr2016-2	4805	173172	0.0000345	1.06e-05	2.0e-07	1.03e-05	1.09e-05	4
Velox	polerr2016-1	13833	135607	0.0001143	3.89e-05	3.0e-07	3.83e-05	3.95e-05	11
Velox	polerr2016-2	5134	45449	0.0001143	4.30e-05	6.0e-07	4.19e-05	4.42e-05	4

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

Error rate consistency between two independent experimental replicas from Polerr2016 dataset.

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

df.er.cast <- merge(df.er.cast, df.er.cast2, by = "name")
colnames(df.er.cast) <- c("name", "replica1.x", "replica2.x", "replica1.y", "replica2.y")

m <- lm(replica1.x ~ replica2.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=replica1.x, y=replica2.x)) +
  geom_errorbarh(aes(xmax=replica1.x+1.96*replica1.y,xmin=replica1.x-1.96*replica1.y)) +
  geom_errorbar(aes(ymax=replica2.x+1.96*replica2.y,ymin=replica2.x-1.96*replica2.y)) +
  geom_smooth(method = "lm", color="black", fill="grey80", fullrange = T) +
  geom_point(size=2, shape=21, fill="grey70") +
  geom_text(aes(label=name), vjust=1, hjust = .3, color="red") +
  geom_label(aes(x = 1e-6, y = 8e-5, label = lbl), hjust=-0.1, parse = TRUE)+
  scale_x_log10(name="Error rate in experiment 1, 1/bp/cycle", limits=c(1e-6,1e-4)) +
  scale_y_log10(name="Error rate in experiment 2, 1/bp/cycle", limits=c(1e-6,1e-4)) +
  theme_bw()
```

```
## Warning: Removed 7 rows containing missing values (geom_smooth).
```



Error substitution patterns

Combine error rates from Polerr2016 replicas, summarize it by substitution type and polymerase.

```
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")
```

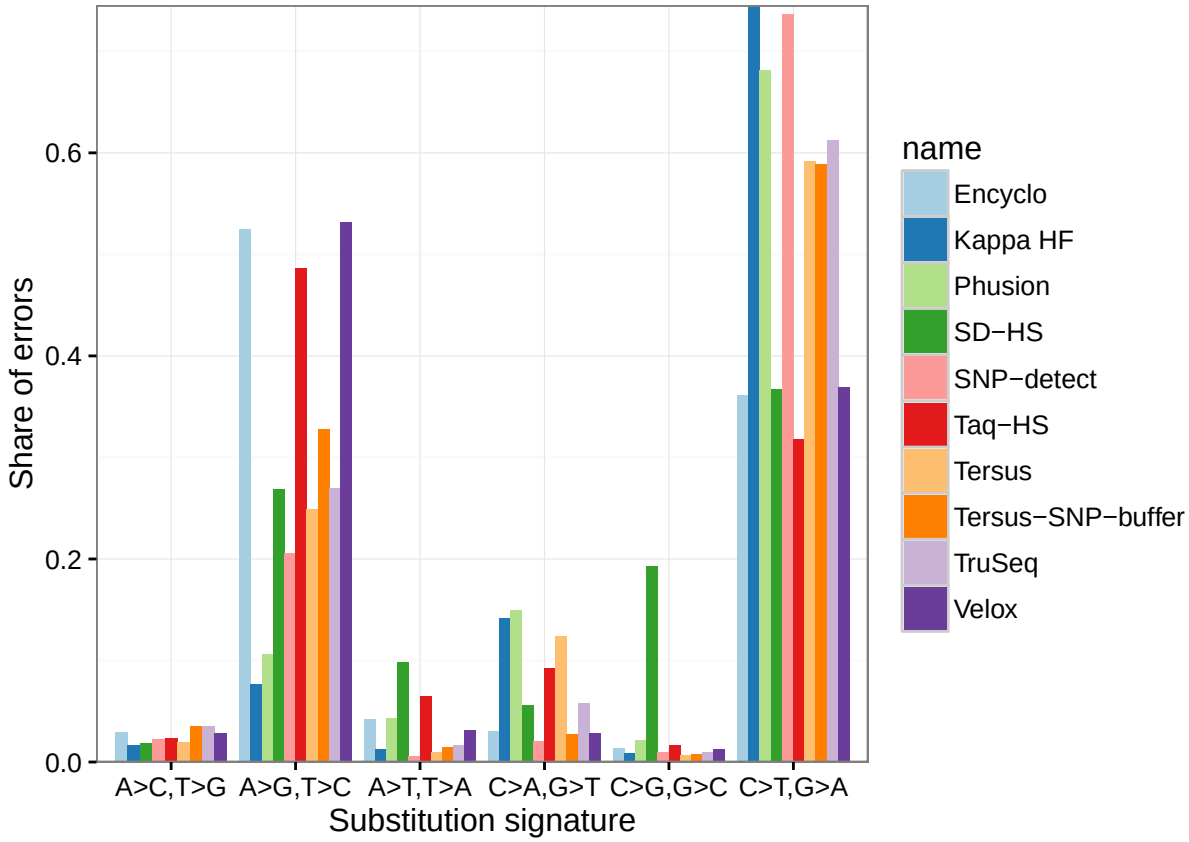
Parse mutation signatures (needed for further analysis)

```
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
```

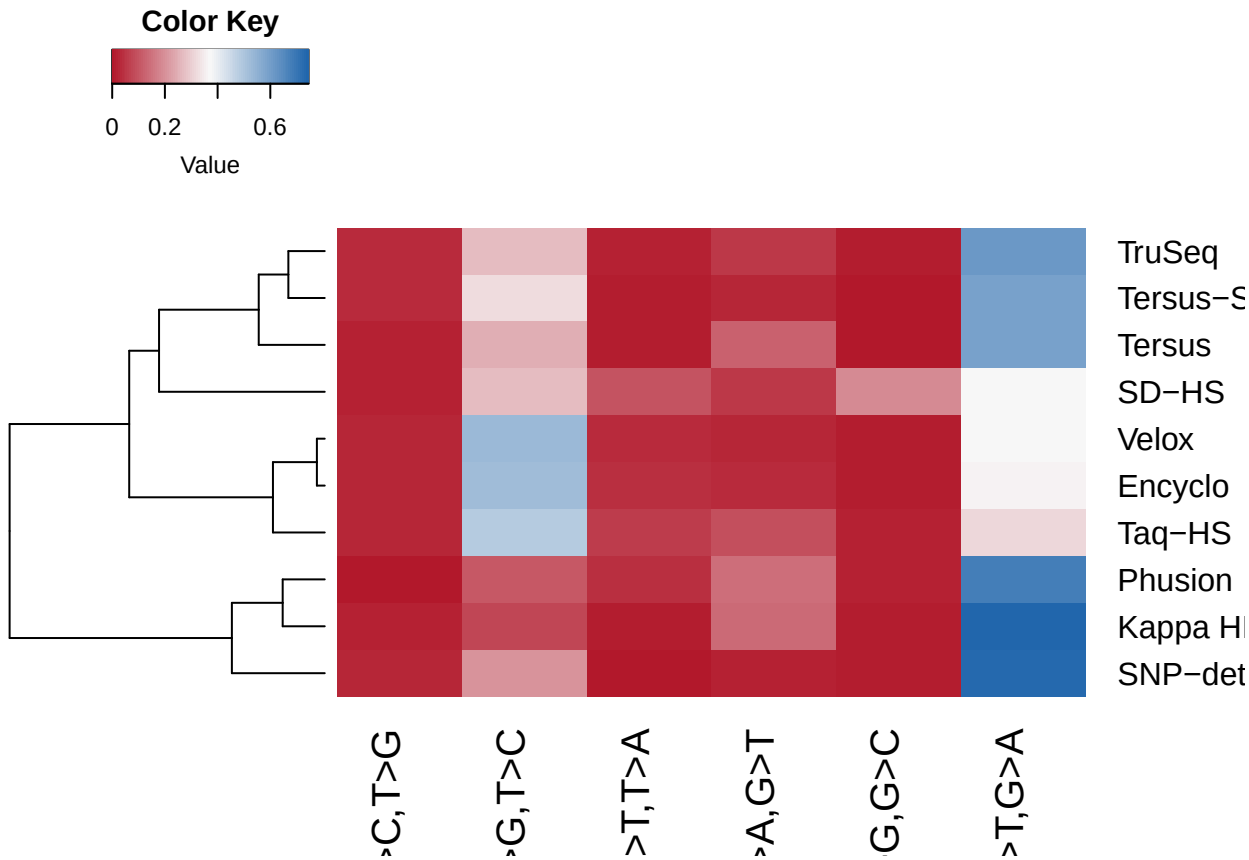
Substitution signature preferences

```
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)
df.pattern.mat[is.na(df.pattern.mat)] <- 0

heatmap.2(df.pattern.mat, col=colorpanel(100, "#b2182b", "#f7f7f7", "#2166ac"),
          dendrogram = "row", Colv=F,
          density.info = "none", trace="none")
```



Transitions and transversions:

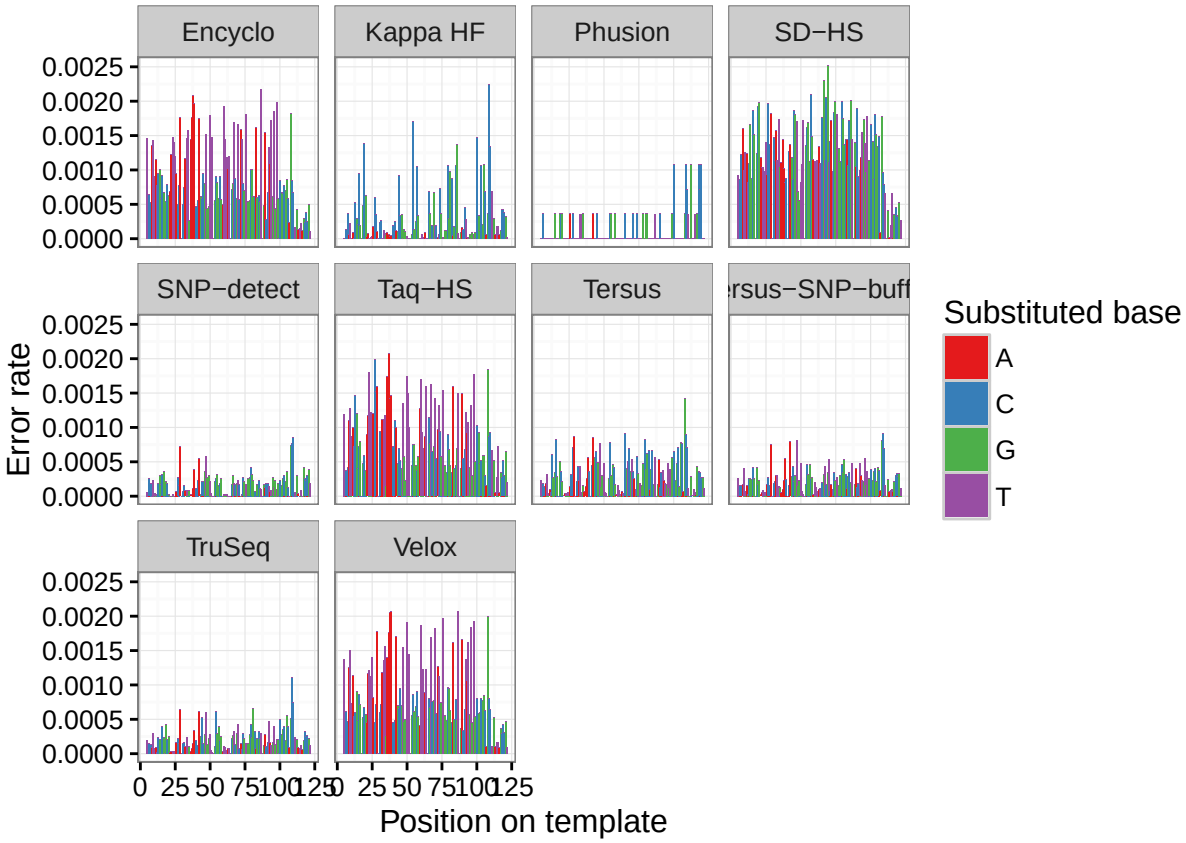
```
x.ti <- rowSums(df.pattern.mat[,c("A>G,T>C","C>T,G>A")])
x.tv <- rowSums(df.pattern.mat) - x.ti

df.trans <- as.data.frame(cbind(x.ti, x.tv))
df.trans$titv <- df.trans$x.ti / df.trans$x.tv
colnames(df.trans) <- c("transitions", "transversions", "Ti/Tv")

kable(df.trans)
```

	transitions	transversions	Ti/Tv
Encyclo	0.8852531	0.1147469	7.714832
Kappa HF	0.8211531	0.1788469	4.591376
Phusion	0.7872340	0.2127660	3.700000
SD-HS	0.6357887	0.3642113	1.745659
SNP-detect	0.9416342	0.0583658	16.133333
Taq-HS	0.8041941	0.1958059	4.107098
Tersus	0.8402094	0.1597906	5.258189
Tersus-SNP-buffer	0.9162541	0.0837459	10.940887
TruSeq	0.8815157	0.1184843	7.439938
Velox	0.9002478	0.0997522	9.024841

This goes to supplementary. Clear preference of substitution patters for different polymerases, but no strong

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()
```



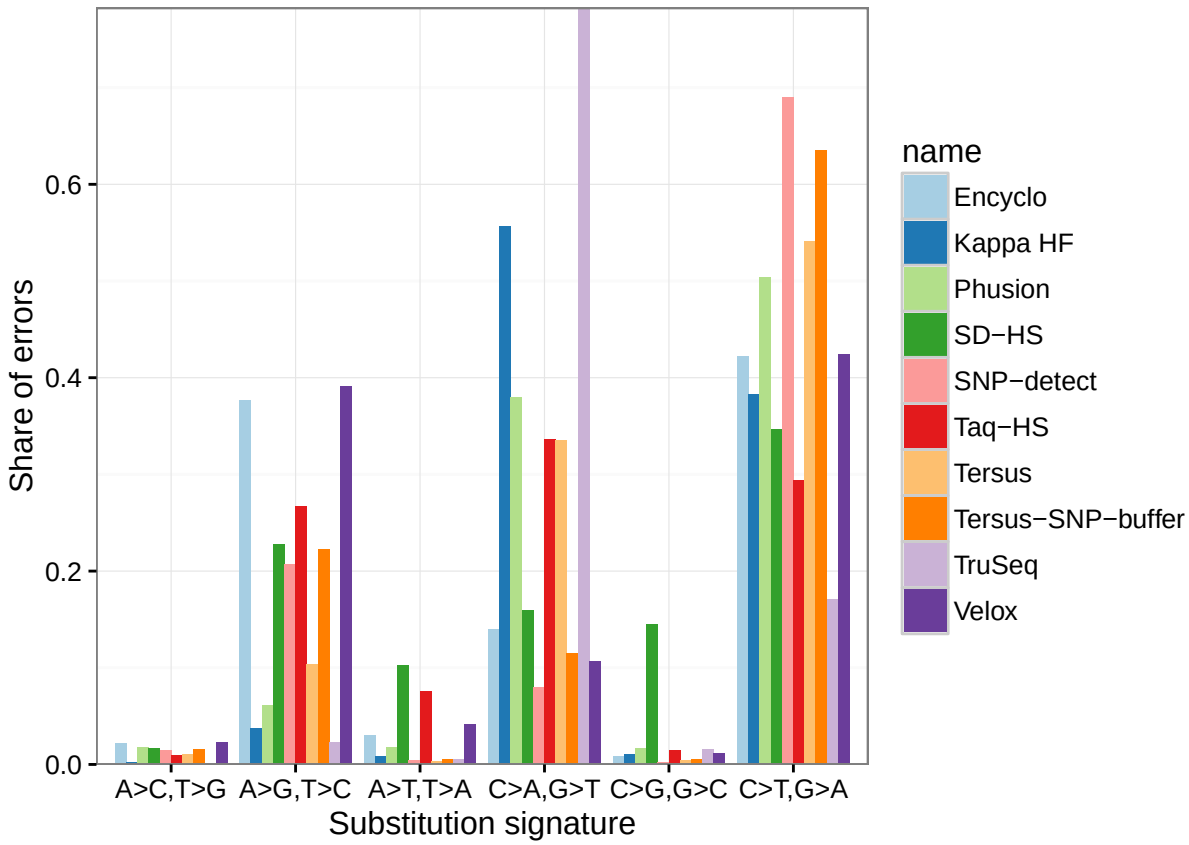

```
a <- aov(count.major / coverage ~ mutation.from * mutation.pos * name, df)
summary(a)
```

```
##                                    Df    Sum Sq   Mean Sq F value   Pr(>F)
## mutation.from                       3 2.800e-07 9.400e-08   0.809   0.4888
## mutation.pos                        1 3.900e-07 3.900e-07   3.342   0.0676
## name                                9 3.103e-05 3.447e-06  29.558  < 2e-16
## mutation.from:mutation.pos          3 1.280e-06 4.260e-07   3.651   0.0121
## mutation.from:name                 27 1.194e-05 4.420e-07   3.792 1.93e-10
## mutation.pos:name                   9 2.230e-06 2.480e-07   2.128   0.0243
## mutation.from:mutation.pos:name    27 8.200e-07 3.100e-08   0.262   1.0000
## Residuals                        2353 2.744e-04 1.170e-07
##
## mutation.from
## mutation.pos                     .
## name                             ***
## mutation.from:mutation.pos       *
## mutation.from:name               ***
```

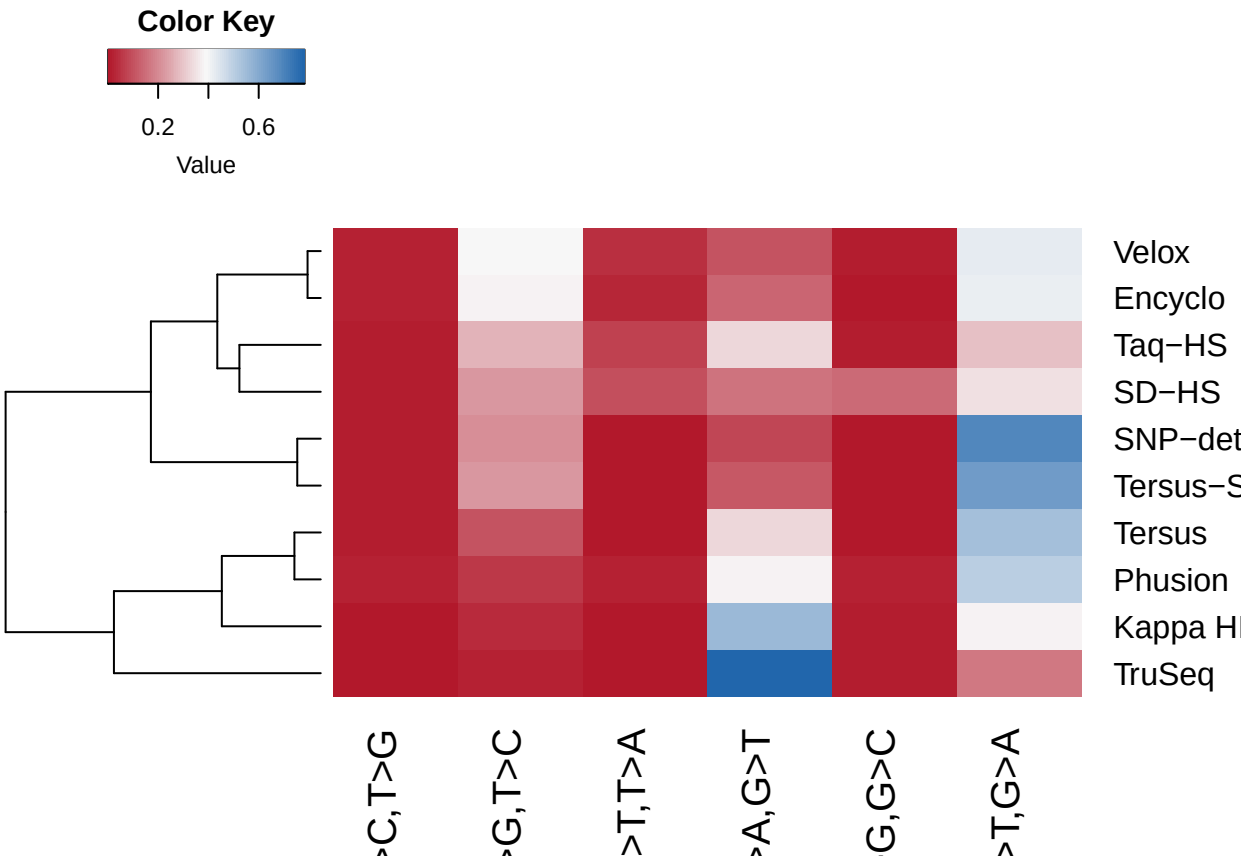
```
## mutation.pos:name               *
## mutation.from:mutation.pos:name
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Same for linear PCR.

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

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

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

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

df.linpcr$mutation.signature <- paste(df.linpcr$mutation.from, df.linpcr$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.linpcr <- merge(df.linpcr, sign.rep, all.x=T, all.y=F)

df.pattern <- ddply(df.linpcr, .(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)
df.pattern.mat[is.na(df.pattern.mat)] <- 0

heatmap.2(df.pattern.mat, col=colorpanel(100, "#b2182b", "#f7f7f7", "#2166ac"),
          dendrogram = "row", Colv=F,
          density.info = "none", trace="none")
```

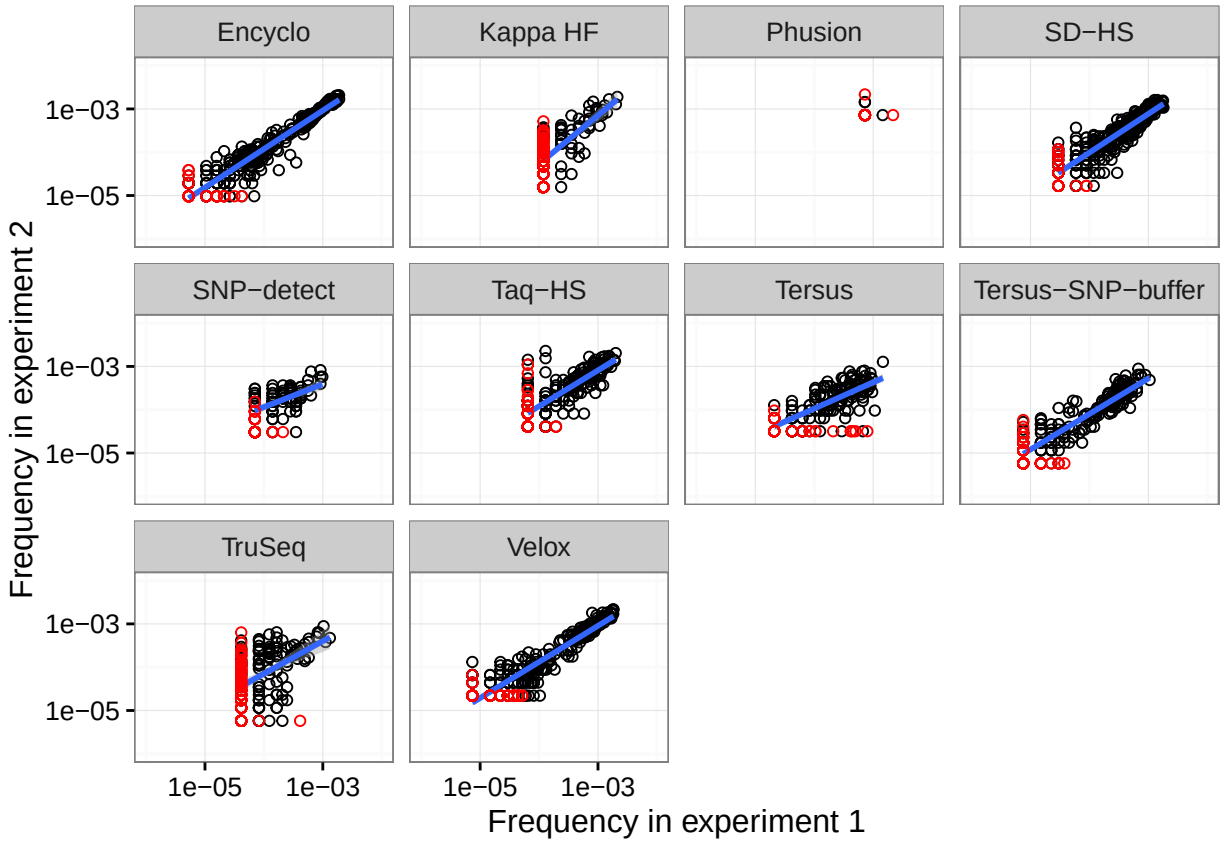



```
x.ti <- rowSums(df.pattern.mat[,c("A>G,T>C","C>T,G>A")])
x.tv <- rowSums(df.pattern.mat) - x.ti

df.trans <- as.data.frame(cbind(x.ti, x.tv))
df.trans$titv <- df.trans$x.ti / df.trans$x.tv
colnames(df.trans) <- c("transitions", "transversions", "Ti/Tv")

kable(df.trans)
```

	transitions	transversions	Ti/Tv
Encyclo	0.7991916	0.2008084	3.9798719
Kappa HF	0.4207807	0.5792193	0.7264618
Phusion	0.5664694	0.4335306	1.3066421
SD-HS	0.5749373	0.4250627	1.3525943
SNP-detect	0.8977778	0.1022222	8.7826087
Taq-HS	0.5620059	0.4379941	1.2831356
Tersus	0.6457772	0.3542228	1.8230822
Tersus-SNP-buffer	0.8584355	0.1415645	6.0639175
TruSeq	0.1945245	0.8054755	0.2415027
Velox	0.8152109	0.1847891	4.4115735

Recurrent errors

```
df.1x <- ddply(subset(df.1, project == "polerr2016-1"), .(name, mutation), summarize,
               freq = count.major / coverage, coverage = coverage)

df.1y <- ddply(subset(df.1, project == "polerr2016-2"), .(name, mutation), summarize,
               freq = count.major / coverage, coverage = coverage)

df.1 <- merge(df.1x, df.1y, by=c("name","mutation"), all = T)
mask1 <- is.na(df.1$freq.x)
mask2 <- is.na(df.1$freq.y)
df.1$miss <- mask1 | mask2

df.1 <- ddply(df.1, .(name), transform,
              freq.x = ifelse(is.na(freq.x), 1/mean(coverage.x, na.rm = T), freq.x),
              freq.y = ifelse(is.na(freq.y), 1/mean(coverage.y, na.rm = T), freq.y))

ggplot() +
  geom_point(data=subset(df.1, !miss), aes(x=freq.x, y=freq.y), shape=21) +
  geom_smooth(data=subset(df.1, !miss & name != "Phusion"), aes(x=freq.x, y=freq.y), method="lm") +
  geom_point(data=subset(df.1, miss), aes(x=freq.x, y=freq.y), shape=21, color="red") +
  facet_wrap(~name) +
  scale_x_log10("Frequency in experiment 1", limits=c(1e-6,1e-2)) +
  scale_y_log10("Frequency in experiment 2", limits=c(1e-6,1e-2)) +
  theme_bw()
```

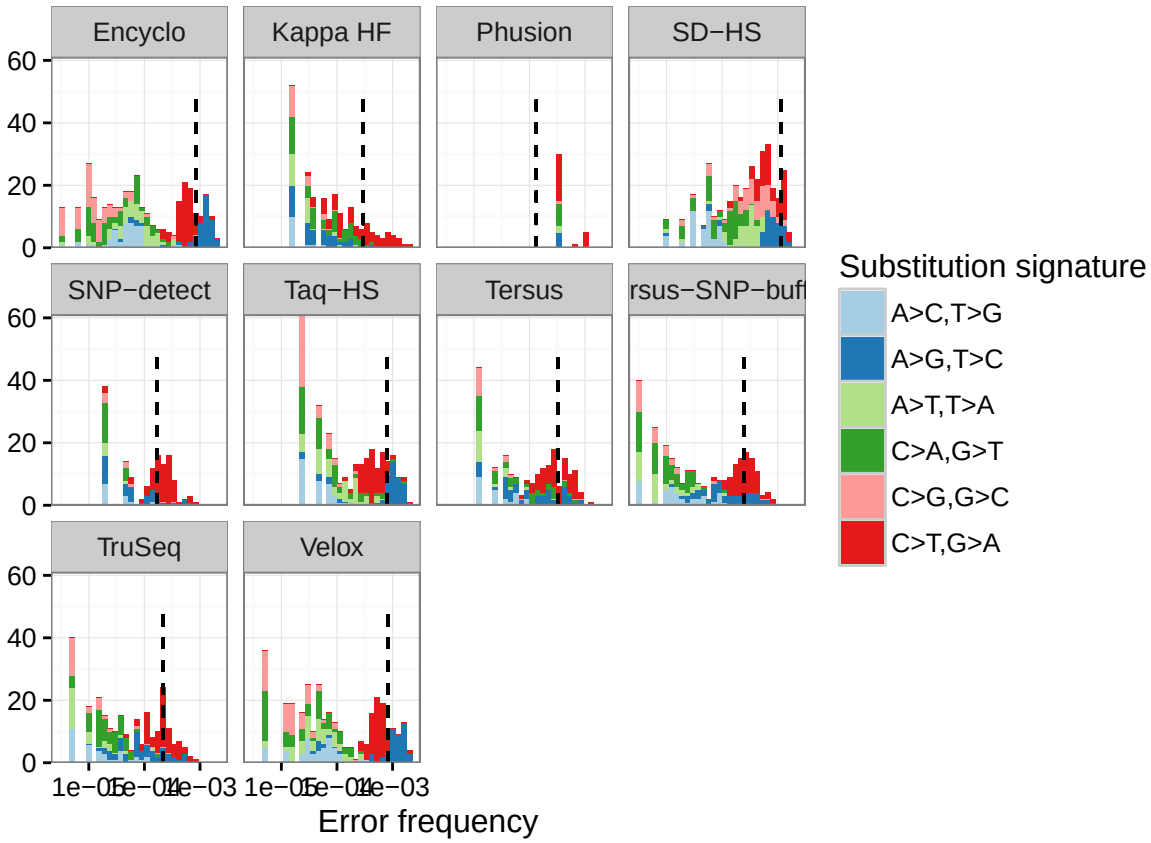


Error hotspot context pattern

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(
# ^ this one is the gloal mean

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()
```

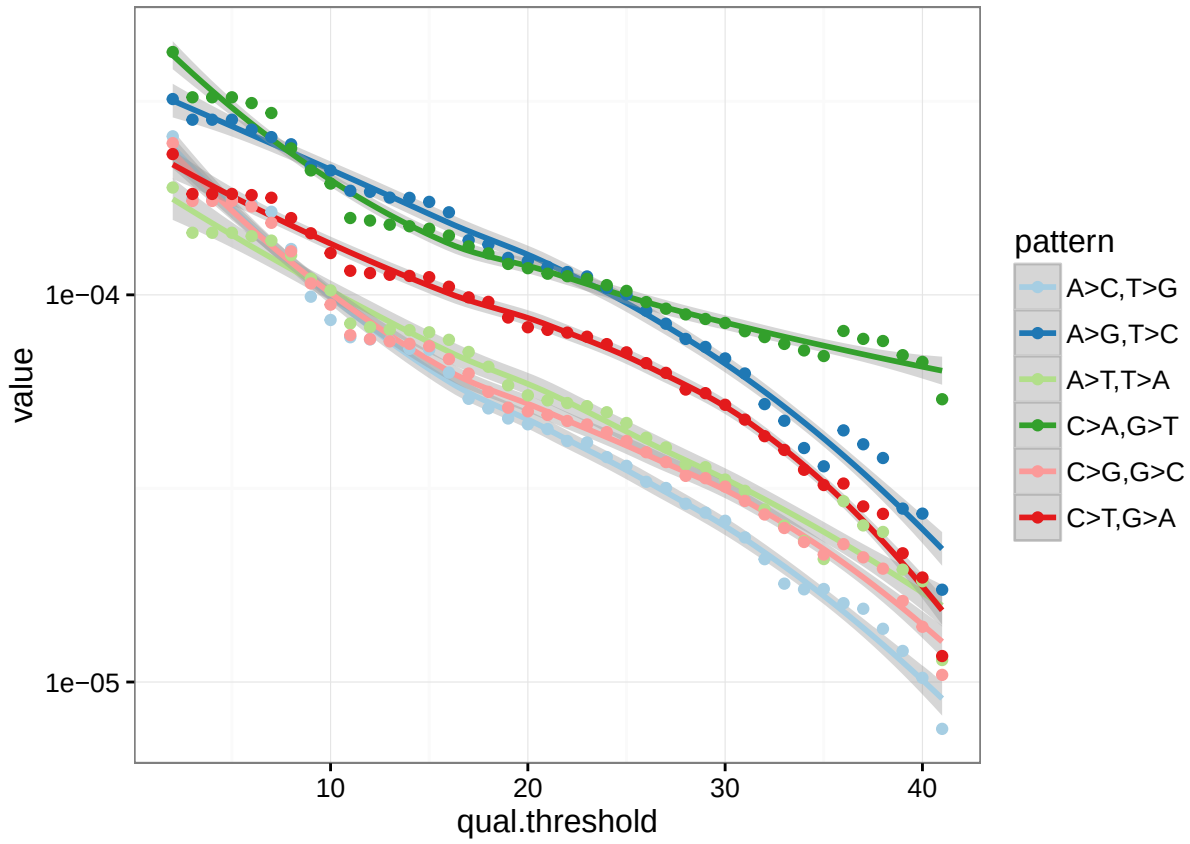
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




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

Sequence quality and error patterns

```
df.q <- read.table("polerr73-82/mmqc.txt", header=T,sep="\t")
library(reshape2)
df.q <- melt(df.q, id.vars = "qual.threshold")

df.q <- merge(df.q,
              data.frame(variable = unlist(strsplit("A.C;T.G;A.G;T.C;A.T;T.A;C.A;G.T;C.G;G.C;C.T;G.A",
                         pattern = unlist(strsplit("A>C,T>G;A>C,T>G;A>G,T>C;A>G,T>C;A>T,T>A;A>T,T>A;C>A
), by = "variable")

df.q <-ddply(df.q, .(qual.threshold, pattern), summarize, value=sum(value))

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

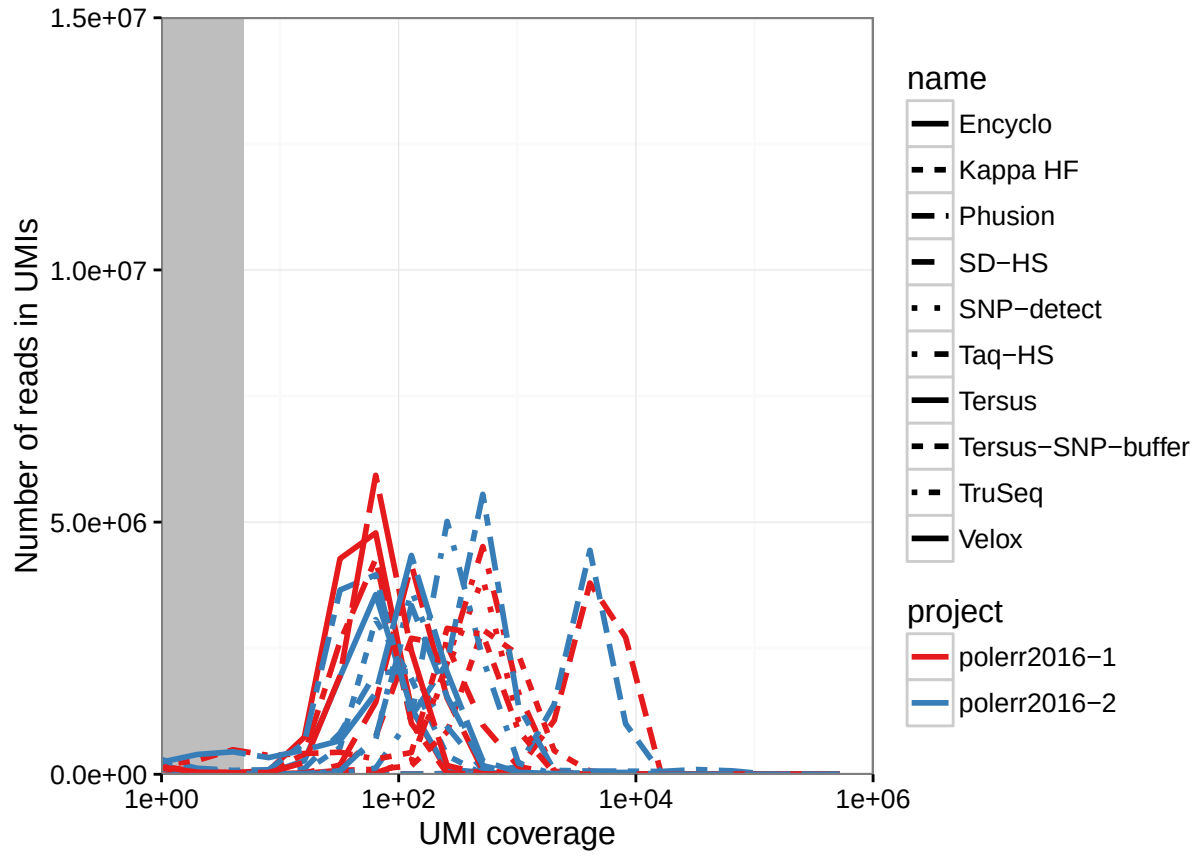


UMI coverage

UMI coverage histogram

```
df.meta <- read.table(paste("polerr2016", "metadata.txt", sep="/"), header=T, sep = "\t")

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(subset(df.meta, project == proj)$sample)) {
    mask <- which(df.meta$sample == sample & df.meta$project == proj)
    name <- df.meta$name[mask][1]
    cycles_2 <- df.meta$cycles_2[mask][1]
    df.hh <- read.table(paste("polerr2016", paste(proj, sample, "umi.histogram.txt", sep="."), sep="/")
                        header=T, sep="\t")
    df.h <- rbind(df.h, data.frame(mig.size.bin = df.hh$mig.size.bin, read.count = df.hh$read.count,
                                   name=name, project=proj, cycles_2=cycles_2))
  }
}


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, cycles_2), summarize,
              peak = log2(mig.size.bin[which(read.count == max(read.count))]))
```

```
m <- lm(peak ~ cycles_2, df.o)
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 = 22, y = 12, label = lbl), hjust=-0.1, parse = TRUE)+
stat_smooth(data=df.o, aes(cycles_2, peak), method=lm, color="black", fill="grey80") +
geom_jitter(data=df.o, aes(cycles_2, peak, fill=name), size=3, width=0.3, height=0.3, color="black", sha
#geom_text(aes(cycles_2,peak,label=name, vjust=1, hjust = .3)) +
scale_x_continuous(name="2nd PCR cycles", breaks=10:30) +
scale_y_continuous(expression('log'[2]~'characteristic MIG size'), breaks=2:20) +
scale_fill_brewer(name  ="Polymerase", palette = "Paired") +
theme_bw()
```