Enzymatic Error Correction Figures

Nathan Lubock

January, 2017

R Stuff

Knitr Options

```
knitr::opts_chunk$set(fig.width = 17.5, fig.height = 10.94, dpi=300)
knitr::opts_chunk$set(fig.path = "./paper/")
knitr::opts_chunk$set(dev='pdf')
knitr::opts_chunk$set(warning=FALSE)

# see http://stackoverflow.com/q/36230790 to scroll output
# needs to be really big to prevent wrapping from happening before the scroll bar comes up
options(width = 240)
```

Initialization

```
# plotting utils
library(scales)
library(grid)
library(gridExtra)

# data.table backend
library(dtplyr)
library(data.table)

# tidyverse!
library(stringr)
library(broom)
library(magrittr)
library(tidyverse)
```

Style Choices

```
theme_pub <- function(base_size = 13, base_family = "") {
  require(grid)
  # based on https://github.com/noamross/noamtools/blob/master/R/theme_nr.R
  # start with theme_bw and modify from there!
  theme_bw(base_size = base_size, base_family = base_family) +# %+replace%
    theme(
      # grid lines
      panel.grid.major.x = element_line(colour="#ECECEC", size=0.5, linetype=1),
```

```
      panel.grid.minor.x = element_blank(),
      panel.grid.minor.y = element_blank(),
      panel.grid.major.y = element_line(colour="#ECECEC", size=0.5, linetype=1),
      panel.background   = element_blank(),

      # axis options
      axis.ticks.y   = element_blank(),
      axis.title.x   = element_text(size=rel(2.25), vjust=0.25),
      axis.title.y   = element_text(size=rel(2.25), vjust=0.35),
      axis.text      = element_text(color="black", size=rel(1.5)),

      # legend options
      legend.title    = element_blank(),
      legend.key      = element_rect(fill="white"),
      legend.key.size = unit(1, "cm"),
      legend.text     = element_text(size=rel(2)),

      # facet options
      strip.text = element_text(size=rel(2)),

      # title options
      plot.title = element_text(size=rel(3), vjust=0.25, hjust=0.5)
      )
  }

# set the theme and brewer color
theme_set(theme_pub())
cbbPalette <- c("#000000", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7")
```

Helper Functions

```
DistribUncert2 <- function(df) {
  # Takes a df with uncertain types and manually distributes them, returning a
  # df of counts ready for plotting. Since we cannot place the insertion or
  # deletion precisely, we will assign fractional counts to each position based
  # on the nature of the difference. For example, a deletion in a 'AAA' repeat
  # could be at any of the A's, so we will assign a 1/3 count to each position
  #
  # Args:
  #   df - a data frame that must have the following:
  #     ColNames: Pos, Diff, Type
  #     Type: Must contain 'UI' and 'UD'
  # Returns:
  #   df - data frame with colnames Pos, Type, Count
  require(dplyr, magrittr, stringr)
  uncert <- df %>%
    filter(Type %in% c('UI', 'UD')) %>%
    mutate(
      To=str_sub(Diff, 2, 2),
      From=str_sub(Diff, 1, 1),
      Diff=str_sub(Diff, -1),
```

```
      Type=str_sub(Type, -1),
      FracCount=1/as.numeric(From)
    ) %>%
    select(c(-From, -To))

  # filter the canonical types and summarize their counts as well
  canon <- df %>%
    filter(!(Type %in% c('UI', 'UD'))) %>%
    mutate(FracCount=1)

  return(bind_rows(uncert, canon))
}

LabelMaker <- function(graph, label){
  # Takes a ggplot and adds a label to the top-left corner.
  # Must be used in conjunction with grid.arrange to plot
  # See: https://stackoverflow.com/a/29863172
  # Args:
  #   graph - a ggplot
  #   label - text string to add as label
  # Returns:
  #   gtable with plot and label
  require(ggplot2, grid, gridExtra)
  myplot <- arrangeGrob(
    graph,
    top = textGrob(
      label,
      x = unit(0, 'npc'),
      y = unit(1, 'npc'),
      just = c('left', 'top'),
      gp = gpar(fontsize=32)
    )
  )

  return(myplot)
}
```

Data Loading

We can get about at 10x speed-up by using pure `data.table`'s, however, some of `dplyr`'s functionality does not seem to behave quite right (especially joins). This is a known issue, and is being worked on.

```
ref.seq <- "GCTGCCGATTTCCATAAGATGCCTCCACGTCTCCGAAGAACTACATGGTGAATGTGTGAAGGCATTTTGAACCAATCCTCGAGCAGTGTTG
refCounts <- data.table(
  Char = c('A', 'T', 'G', 'C', 'N'),
  Count = c(str_count(ref.seq, 'A'),
            str_count(ref.seq, 'T'),
            str_count(ref.seq, 'G'),
            str_count(ref.seq, 'C'),
            str_count(ref.seq, 'N'))
  )

# requisite information for all treatments
```

```
# setwd("~/Projects/errorCorrect-dev/analysis/")
charCounts <- fread('zcat ./output/char-counts.txt.gz', header=T)
data <- fread('zcat ./output/errs-all-samples.csv.gz', header=T)
allSamps <- data %>%
  filter(str_sub(Sample, 1, 1) %in% c('1', '2'))

# setwd("~/Dropbox/UCLA/Kosuri/ErrorCorrect/github/errorCorrect/analysis/")
# charCounts <- fread('./data/output/char-counts.txt', header=T)
# data <- fread('./data/output/errs-all-samples.csv', header=T)
# allSamps <- data %>%
#   filter(str_sub(Sample, 1, 1) %in% c('1', '2'))

# constants for all samples
readCounts <- fread('./output/read-counts.txt', header=T)

# subset variables for easy running
nonDoped <- allSamps %>% filter(Sample == '1_nonDoped')
doped <- allSamps %>% filter(Sample == '1_DopedTemp')
```

Review Setup

```
c2 <- 'GGGTCACGCGTAGGACATTACTCTACGGTAAGGCGACATATGGCCAGATTAATTCCCAGTGTCGTCTAAGCGGTCTATCATAAATCGTGGATGGAG
c3 <- 'GGTCGAGCCGGAACTGATGACGATGGCTAAATACATGTATAATGGTACATCCCTAGCAGGCATGCCCGCTTGGTCACGTGAGTAGCCCACGATATC

review.samples <- c(
  'C2-Q5-1', 'C2-Q5-2', 'C3-Q5-1', 'C3-Q5-2',
  'C2-Taq-1', 'C2-Taq-2', 'C3-Taq-1', 'C3-Taq-2')

review.counts <- refCounts <- data.table(
  Char = rep(c('A', 'T', 'G', 'C'), 2),
  Construct = c(rep('C2', 4), rep('C3', 4)),
  Count = c(str_count(c2, 'A'), str_count(c2, 'T'), str_count(c2, 'G'), str_count(c2, 'C'),
            str_count(c3, 'A'), str_count(c3, 'T'), str_count(c3, 'G'), str_count(c3, 'C'))
)

review <- data %>%
  filter(Sample %in% review.samples)

# free up ram
rm(data)
```

Main Figures

Figure 2 - Error Analysis for a Standard Oligo Assembly

```
#-------------------------------------------------------------------------------
# Panel 1
# Plot the position of all types of errors
```

```r
# We need to make sure that any 0's are actually caught for plotting
# nonDoped %>%
#   DistribAndNorm(., 1) %>%
#   complete(Type, Pos, fill=list('Norm'=0))

positions <- nonDoped %>%
  DistribUncert2() %>%
  count(Pos, Type, wt=FracCount) %>%
  filter(Type != 'S') %>%
  ungroup() %>%
  mutate(
    Norm = n / subset(readCounts, Sample == '1_nonDoped')$Reads,
    Type = Type %>%
      factor(levels = c('M', 'I', 'D', 'P')) %>%
      recode(D = 'Single Base Deletions', I = 'Single Base Insertions',
             P = 'Multiple Base Deletions', M = 'Mismatches')
  ) %>%
  ggplot(aes(x=Pos, y=Norm)) +
  geom_point(size=3) +
  facet_wrap(~ Type, ncol=2) +
  stat_smooth(se=F) +
  labs(x = 'Position',
       y = 'Error Rate',
       title = 'Error Rate vs. Position for Error Sub-types') +
  scale_y_log10() +
  annotation_logticks(sides='l')

#-------------------------------------------------------------------------------
# Panel 1b
# Percentage of Error Subtypes

sub_type <- nonDoped %>%
    DistribUncert2() %>%
  count(Type, wt=FracCount) %>%
  mutate(
    Norm = n / sum(n) * 100,
    Type = Type %>%
      factor(levels = c('M', 'D', 'P', 'I', 'S')) %>%
      recode(M = 'MM', D = 'Del.', P = 'M. Del.', I = 'Ins.', S = 'M. Ins.')
    ) %T>%
  {arrange(., -Norm) %>% print()} %>%
  ggplot(aes(x=Type, y=Norm)) +
  geom_bar(stat='identity') +
  theme(plot.title = element_text(size=rel(2))) +
  labs(
    y = 'Percent',
    x = 'Error Type',
    title = 'Percentage of Error Sub-types'
  )
```

```
## Source: local data table [5 x 3]
##
## # tbl_dt [5 × 3]
```

```
##      Type      n       Norm
##    <fctr>  <dbl>      <dbl>
## 1      MM 155254 75.0682971
## 2    Del.  29313 14.1733997
## 3 M. Del.  16946  8.1937172
## 4    Ins.   4897  2.3677938
## 5 M. Ins.    407  0.1967923
```

```
#-------------------------------------------------------------------------------
# Panel 1c
# plot the distribution of total mismatches per position
mm_freq <- nonDoped %>%
  filter(Type == 'M') %>%
  mutate(
    To = str_sub(Diff, 2, 2),
    From = str_sub(Diff, 1, 1)
  ) %>%
  count(Pos, From) %>%
  ungroup() %>%
  # left_join(rename(refCounts, From=Char), by='From') %>%
  mutate(Norm = n / subset(readCounts, Sample == '1_nonDoped')$Reads) %T>%
  { # pairwise wilcox test and print median values for paper
    group_by(., From) %>%
      summarise(med=median(Norm)) %>%
      arrange(-med) %>%
      print; # <- ; critical for . to be interpreted correctly
    with(., pairwise.wilcox.test(n, From)) %>% print
  } %>%
  ggplot(aes(x = From, y = Norm)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(position=position_jitter(w = 0.5), size=0.75, alpha=0.8) +
  # stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median, geom = 'crossbar', width = 0.5) +
  labs(
    y = 'Error Rate',
    title = 'Mismatches per Position'
  ) +
  theme(
    axis.text.x = element_text(size=28),
    axis.title.x = element_blank(),
    plot.title = element_text(size=rel(2.25))
  ) +
  scale_y_continuous(labels = scientific_format())
```

```
## Source: local data table [4 x 2]
##
## # tbl_dt [4 × 2]
##    From         med
##   <chr>       <dbl>
## 1     A 0.004337145
## 2     T 0.004247793
## 3     C 0.001912329
## 4     G 0.001682274
##
##  Pairwise comparisons using Wilcoxon rank sum test
```

```
##
## data:  n and From
##
##   A       C       G
## C 3.3e-12 -       -
## G 7.2e-08 0.47    -
## T 0.58    1.2e-12 5.7e-08
##
## P value adjustment method: holm
```

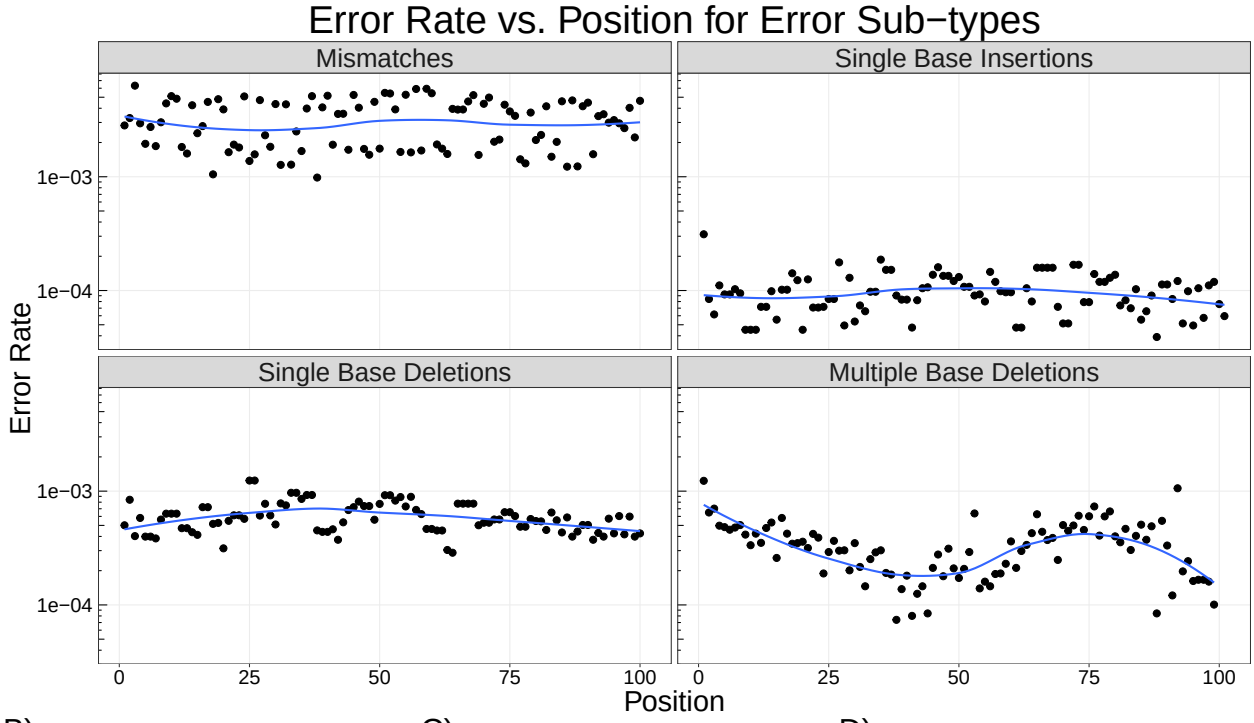
```
#-------------------------------------------------------------------------------
# Panel 1d
# what bases are most likely mutated to
# We will normallize by the total count in each "from" group
mm_type <- nonDoped %>%
  filter(Type == 'M') %>%
  count(Pos, Diff) %>%
  ungroup() %>%
  mutate(
    Char=str_sub(Diff, 1, 1),
    Class=Diff %>%
      recode(AT='Transversion', AG='Transition', AC='Transversion',
             TA='Transversion', TG='Transversion', TC='Transition',
             GA='Transition', GT='Transversion', GC='Transversion',
             CA='Transversion', CT='Transition', CG='Transversion')
  ) %>%
  # left_join(refCounts, by='Char') %>%
  mutate(Norm = n / subset(readCounts, Sample == '1_nonDoped')$Reads) %T>%#(Count * subset(readCounts,
  {# significance testing and printing for paper
    group_by(., Diff) %>%
      summarise(med=median(Norm)) %>%
      arrange(-med) %>%
      print; # <- ; critical for . to be interpreted correctly
      with(., pairwise.wilcox.test(Norm, Diff)) %>% print
  } %>%
  ggplot(aes(x = Diff, y = Norm, color=Class)) +
  geom_boxplot(outlier.shape = NA, show.legend = FALSE) +
  geom_jitter(position=position_jitter(w = 0.5), size=0.75, alpha=0.8) +
  # stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median, geom = 'crossbar', width = 0.5,
  labs(
    y = 'Error Rate',
    title = 'Mismatch Sub-types per Pos.'
    ) +
  theme(
    legend.position='bottom',
    legend.key.size=unit(0.75, "cm"),
    axis.title.x=element_blank(),
    axis.text.x = element_text(angle = 315, vjust=0.5),
    plot.title = element_text(size=rel(1.75))
  ) +
  scale_y_continuous(labels = scientific_format()) +
  scale_color_manual(values = c('#7b3294', '#008837')) +
  guides(colour = guide_legend(override.aes = list(size=5)))
```

```
## Source: local data table [12 x 2]
##
## # tbl_dt [12 × 2]
##     Diff          med
##    <chr>        <dbl>
## 1     TC 0.0034939539
## 2     AG 0.0034128186
## 3     CT 0.0012776245
## 4     GA 0.0012406515
## 5     AT 0.0006850286
## 6     TA 0.0005925959
## 7     CG 0.0002957845
## 8     GT 0.0002896223
## 9     CA 0.0002218383
## 10    GC 0.0001910275
## 11    AC 0.0001540544
## 12    TG 0.0001427571
##
##  Pairwise comparisons using Wilcoxon rank sum test
##
## data:  Norm and Diff
##
##    AC      AG      AT      CA      CG      CT      GA      GC      GT      TA      TC
## AG 2.9e-08 -       -       -       -       -       -       -       -       -       -
## AT 4.0e-08 2.5e-13 -       -       -       -       -       -       -       -       -
## CA 0.0134  2.0e-12 8.1e-06 -       -       -       -       -       -       -       -
## CG 0.2814  2.0e-12 1.7e-05 1.0000  -       -       -       -       -       -       -
## CT 8.4e-08 2.0e-12 1.1e-06 5.1e-11 1.3e-11 -       -       -       -       -       -
## GA 8.4e-08 5.3e-11 2.8e-05 8.7e-11 2.6e-11 1.0000  -       -       -       -       -
## GC 0.4843  1.3e-07 1.3e-05 1.0000  1.0000  3.2e-06 3.1e-06 -       -       -       -
## GT 0.0248  8.4e-08 8.7e-06 1.0000  1.0000  2.3e-07 2.3e-07 1.0000  -       -       -
## TA 2.0e-08 6.9e-14 1.0000  4.0e-07 5.3e-05 5.4e-07 3.1e-06 8.1e-06 4.5e-06 -       -
## TC 1.5e-08 1.0000  6.9e-14 6.3e-13 6.3e-13 1.2e-12 6.8e-12 6.4e-08 4.9e-08 1.7e-14 -
## TG 1.0000  1.5e-08 2.7e-08 0.0077  0.2857  4.9e-08 4.9e-08 0.4373  0.0220  1.4e-08 7.0e-09
##
## P value adjustment method: holm
```

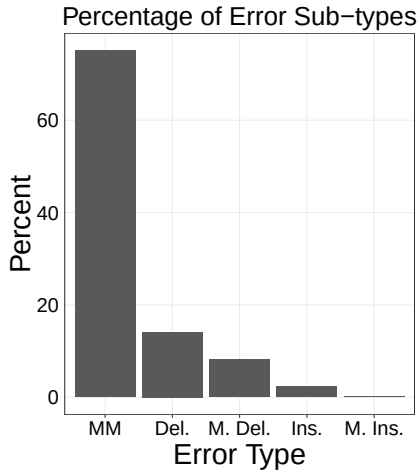
```
#-------------------------------------------------------------------------
# plot everything!
grid.arrange(
  LabelMaker(positions, 'A)'),
  arrangeGrob(
    LabelMaker(sub_type, 'B)'),
    LabelMaker(mm_freq, 'C)'),
    LabelMaker(mm_type, 'D)'),
    nrow=1),
  ncol=1,
  heights = c(1, 0.67)
  )
```

```
## `geom_smooth()` using method = 'loess'
```

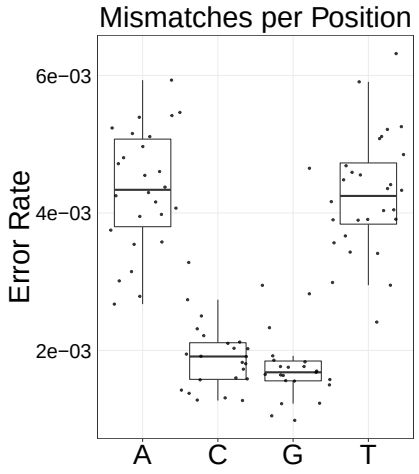
A)



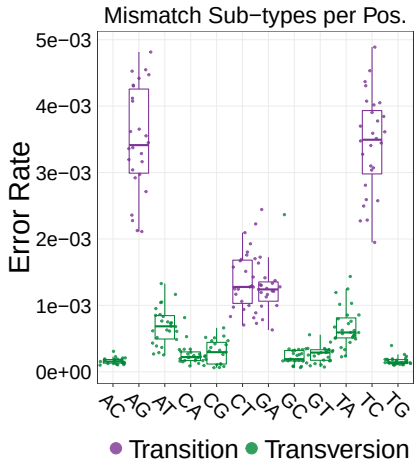
B)



C)



D)




```
# pairwise test of medians for each group
nonDoped %>%
  DistribUncert2() %>%
  count(Type, Pos, wt=FracCount) %>%
  mutate(Norm = n / subset(readCounts, Sample == '1_nonDoped')$Reads) %T>%
  with(., pairwise.wilcox.test(Norm, Type) %>% print) %>%
  group_by(Type) %>%
  summarise(med=median(Norm), mean=mean(Norm))
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test
##
## data:  Norm and Type
```

```
##
##   D      I      M      P
## I <2e-16 -      -      -
## M <2e-16 <2e-16 -      -
## P <2e-16 <2e-16 <2e-16 -
## S <2e-16 <2e-16 <2e-16 <2e-16
##
## P value adjustment method: holm
```

```
## Source: local data table [5 x 3]
##
## # tbl_dt [5 × 3]
##    Type          med         mean
##   <chr>        <dbl>        <dbl>
## 1     D 5.638391e-04 6.021062e-04
## 2     I 9.654076e-05 9.959134e-05
## 3     M 3.079034e-03 3.189008e-03
## 4     P 3.348116e-04 3.515968e-04
## 5     S 6.162176e-06 8.277247e-06
```

```
# insertions at position 1
nonDoped %>%
  DistribUncert2() %>%
  filter(Type == 'I', Pos == 1) %>%
  count(Diff)
```

```
## Source: local data table [4 x 2]
##
## # tbl_dt [4 × 2]
##    Diff     n
##   <chr> <int>
## 1     T    57
## 2     C    16
## 3     G    78
## 4     A     1
```

```
# differences in medians in annealing regions and outside
nonDoped %>%
    DistribUncert2() %>%
    count(Type, Pos, wt=FracCount) %>%
    mutate(Norm = n / subset(readCounts, Sample == '1_nonDoped')$Reads) %>%
    filter(Type == 'P') %>%
  mutate(Region = if_else(Pos >= 36 & Pos <=64, 'Anneal', 'No')) %T>%
  {wilcox.test(Norm ~ Region, data=.) %>% print} %>%
  group_by(Region) %>%
  summarise(med=median(Norm), IQR=IQR(Norm))
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Norm by Region
## W = 371, p-value = 7.507e-07
## alternative hypothesis: true location shift is not equal to 0
```

```
## Source: local data table [2 x 3]
##
## # tbl_dt [2 × 3]
##   Region          med          IQR
##    <chr>        <dbl>        <dbl>
## 1     No 0.0003892441 0.0002018113
## 2 Anneal 0.0001889734 0.0001314598
```

Figure 3 - Error Correction and Percent Perfects

```
# reduce the all of the reads down to errors/read (this may take a while...)
# join with the length of every read and fill in perfects with a 0 count
# errRate calculation from Furhman paper
errRates <- allSamps %>%
  DistribUncert2() %>%
  count(Sample, Name, wt=FracCount) %>%
  left_join(charCounts %>% filter(!Sample %in% review.samples), ., by = c('Sample', 'Name')) %>%
  replace_na(list(n=0)) %>%
  group_by(Sample) %>%
  summarise(
    errRate = mean(n * (1000/Len)),
    sem = sd(n*(1000/Len)) / sqrt(n())
  ) %>%
  left_join(readCounts, by='Sample') %>%
  mutate(
    PercentPerf = (Reads - Errs) / Reads * 100,
    Treatment = str_sub(Sample, 1, 1),
    Sample = str_sub(Sample, 3),
    # Pretty printing for figures
    Sample = Sample %>%
      recode(nonDoped='Standard Oligo', DopedTemp='Doped Oligo',
             MutS_1900nM='MutS (1900nM)', MutS_950nM='MutS (950nM)',
             T7EndoIFurhmann='T7 EndoI (Fuhrmann)', T4EndoVII='T4 EndoVII',
             T7EndoI='T7 EndoI (0U T7 Lig.)',
             `T7EndoI-e3T7Ligase`='T7 EndoI (1e3U T7 Lig.)',
             `T7EndoI-e4T7Ligase`='T7 EndoI (1e4U T7 Lig.)',
             `ErrASE-nonDoped` = 'Standard Oligo (ErrASE)')
  )
```

```
# manual ordering for nice plots
plt.order <- c('Standard Oligo (ErrASE)', 'Standard Oligo', 'MutS (1900nM)',
           'MutS (950nM)', 'ErrASE', 'T7 EndoI (Fuhrmann)',
           'T4 EndoVII', 'Surveyor', 'T7 EndoI (0U T7 Lig.)',
           'T7 EndoI (1e3U T7 Lig.)', 'T7 EndoI (1e4U T7 Lig.)',
           'EndoV', 'Doped Oligo')

errRates %>%
  # add in 0's for doped and nonDoped
  select(c(-Errs, -Reads, -sem)) %>%
  bind_rows(
    data.table(Sample=c('Doped Oligo', 'Standard Oligo'),
               errRate=c(0.0, 0.0),
```

```
              PercentPerf=c(0.0, 0.0),
              Treatment=c('2', '2'))
) %>%
gather(Metric, Value, PercentPerf, errRate) %>%
mutate(
  Metric=if_else(Metric == 'PercentPerf',
                 "Percent Perfect Reads",
                 "Error Frequency (per kb)") %>%
    factor(levels=c("Percent Perfect Reads",
                    "Error Frequency (per kb)")),
  Sample = factor(Sample, levels = plt.order)
) %>%
ggplot(aes(x=Sample, y=Value, fill=Treatment)) +
geom_bar(stat='identity', position='dodge') +
facet_wrap(~ Metric, nrow=2) +
theme(
  axis.text.x=element_text(angle=315, hjust=0.15, vjust=0.90, size=rel(1.0)),
  axis.title.x=element_blank(),
  axis.title.y=element_blank(),
  legend.title=element_text(size=rel(2.25)),
  legend.position="bottom"
  ) +
scale_y_continuous(breaks=seq(0,60,10)) +
scale_fill_manual(name="Treatment Round:",
                  values=c("#ca0020", "#0571b0"))
```

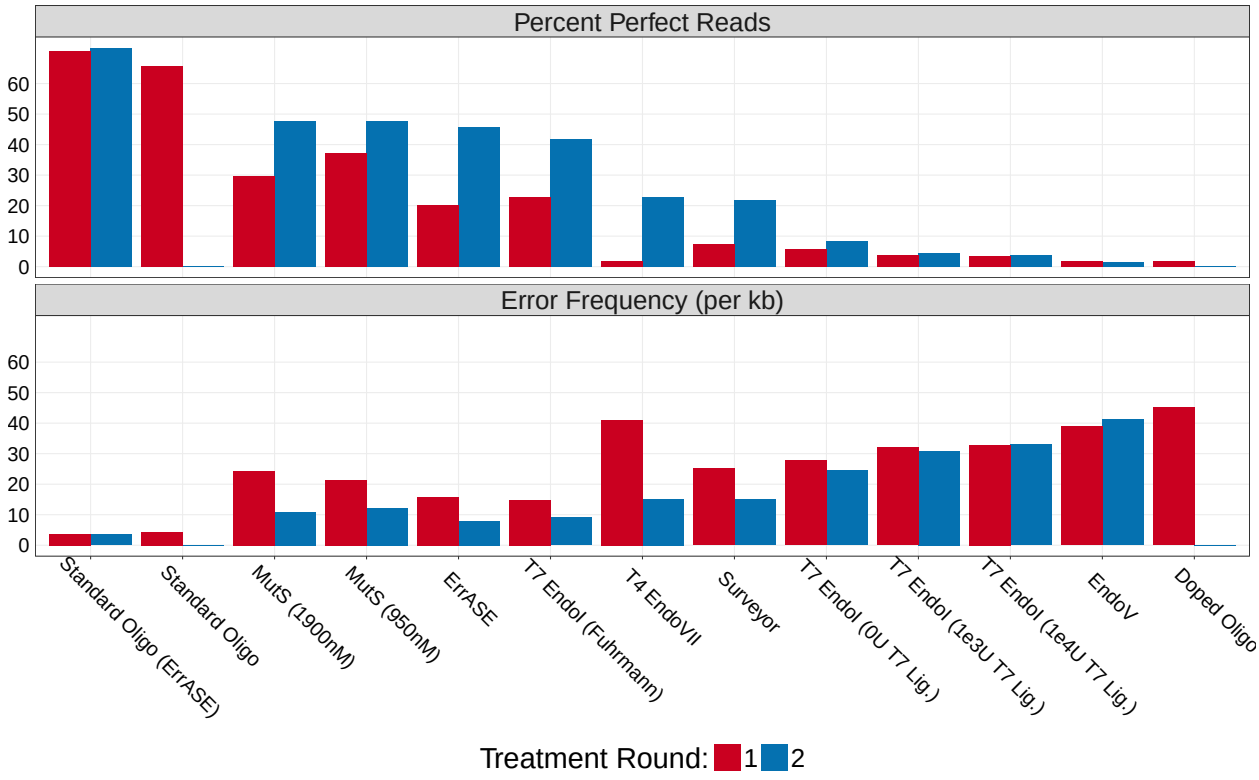


Figure 4 - Enzyme Preferences

```
enzPref <- allSamps %>%
  DistribUncert2() %>%
  count(Sample, Type, Pos, wt=FracCount) %>%
  ungroup() %>%
  left_join(readCounts, by = 'Sample') %>%
  mutate(
    Treatment=str_sub(Sample, 1, 1),
    Sample=str_sub(Sample, 3),
    Norm=n / Reads
  ) %>%
  # Grab the normalized data from DopedTemp for easy divide
  left_join(.,
            filter(., Sample == 'DopedTemp') %>%
              transmute(Type=Type, Pos=Pos, Dope_Norm=Norm),
            by = c('Type', 'Pos')
  ) %>%
  mutate(
    Rel_Norm = Norm / Dope_Norm,
    Fold_2 = log2(Rel_Norm),
    Fold = 2^-Fold_2,
    # pretty print for figures
    Sample = Sample %>%
      recode(nonDoped='Standard Oligo', DopedTemp='Doped Oligo',
             MutS_1900nM='MutS (1900nM)', MutS_950nM='MutS (950nM)',
             T7EndoIFurhmann='T7 EndoI (Fuhrmann)', T4EndoVII='T4 EndoVII',
             T7EndoI='T7 EndoI (0U T7 Lig.)',
             `T7EndoI-e3T7Ligase`='T7 EndoI (1e3U T7 Lig.)',
             `T7EndoI-e4T7Ligase`='T7 EndoI (1e4U T7 Lig.)',
             `ErrASE-nonDoped` = 'Standard Oligo (ErrASE)')
  )

#-------------------------------------------------------------------------------
# specific call-outs for indels and mismatches
enzPref.idm <- allSamps %>%
  DistribUncert2() %>%
  filter(Type %in% c('I', 'M', 'D')) %>%
  count(Sample, Type, Pos, Diff, wt=FracCount) %>%
  ungroup() %>%
  left_join(readCounts, by = 'Sample') %>%
  mutate(
    Treatment=str_sub(Sample, 1, 1),
    Sample=str_sub(Sample, 3),
    Norm=n / Reads
    ) %>%
  # Grab the normalized data from DopedTemp for easy divide
  left_join(.,
            filter(., Sample == 'DopedTemp') %>%
              transmute(Diff=Diff, Type=Type, Pos=Pos, Dope_Norm=Norm),
            by = c('Type', 'Pos', 'Diff')
            ) %>%
  mutate(
```

```
    Rel_Norm = Norm / Dope_Norm,
    Fold_2 = log2(Rel_Norm),
    Fold = 2^-Fold_2,
    Class= Diff %>%
      recode(AT='Transversion', AG='Transition', AC='Transversion',
             TA='Transversion', TG='Transversion', TC='Transition',
             GA='Transition', GT='Transversion', GC='Transversion',
             CA='Transversion', CT='Transition', CG='Transversion'),
    Sample = Sample %>%
      recode(nonDoped='Standard Oligo', DopedTemp='Doped Oligo',
             MutS_1900nM='MutS (1900nM)', MutS_950nM='MutS (950nM)',
             T7EndoIFurhmann='T7 EndoI (Fuhrmann)', T4EndoVII='T4 EndoVII',
             T7EndoI='T7 EndoI (0U T7 Lig.)',
             `T7EndoI-e3T7Ligase`='T7 EndoI (1e3U T7 Lig.)',
             `T7EndoI-e4T7Ligase`='T7 EndoI (1e4U T7 Lig.)',
             `ErrASE-nonDoped` = 'Standard Oligo (ErrASE)')
  )
```

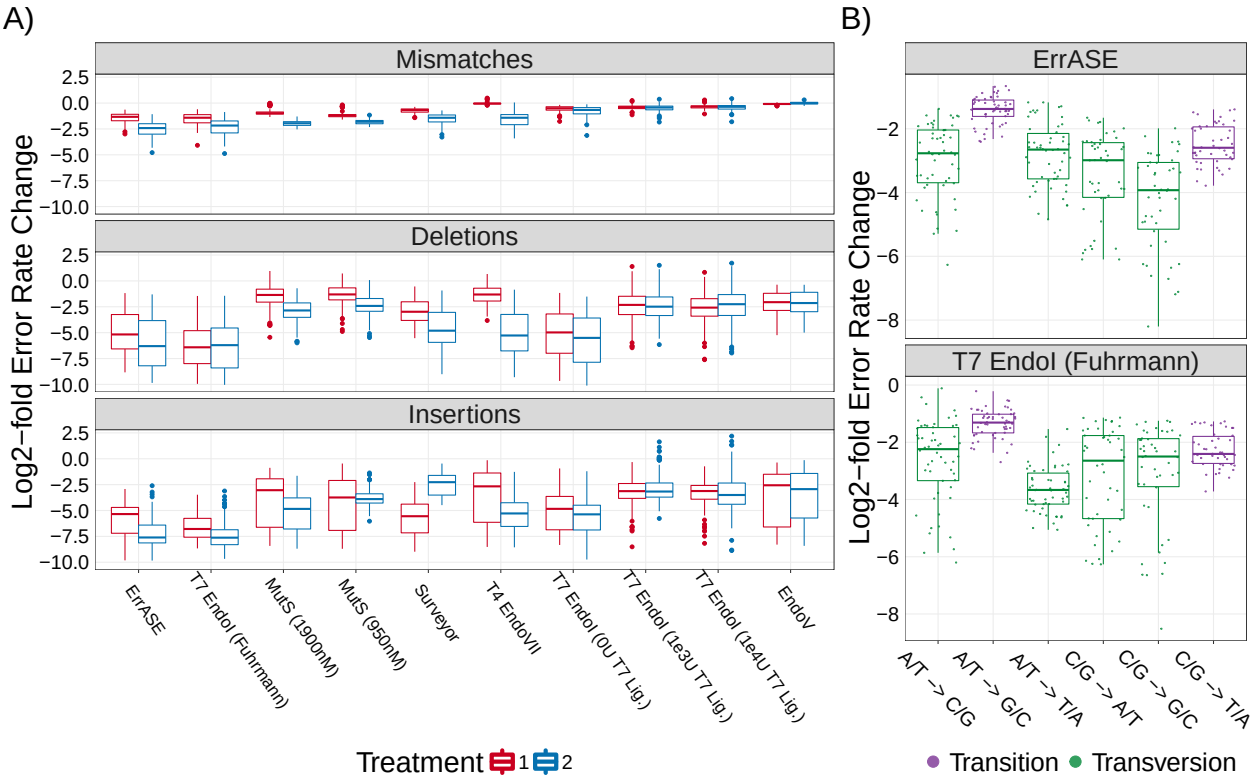
```
# order by error frequency
plt.order <- c('Standard Oligo (ErrASE)', 'Standard Oligo',
               'ErrASE', 'T7 EndoI (Fuhrmann)', 'MutS (1900nM)',
               'MutS (950nM)', 'Surveyor', 'T4 EndoVII',
               'T7 EndoI (0U T7 Lig.)', 'T7 EndoI (1e3U T7 Lig.)',
               'T7 EndoI (1e4U T7 Lig.)', 'EndoV', 'Doped Oligo')

#-------------------------------------------------------------------------------
# Panel 1
# Plot the positional distribution across enzymes for indels and mm's
pan1 <- enzPref %>%
  filter(!Sample %in% c('Doped Oligo', 'Standard Oligo', 'Standard Oligo (ErrASE)')) %>%
  mutate(
    Type = case_when(Type == 'D' | Type == 'P' ~ 'Deletions',
                     Type == 'I' | Type == 'S' ~ 'Insertions',
                     TRUE ~ 'Mismatches'),
    # factor madness for proper ordering
    Type = factor(Type, levels = c('Mismatches', 'Deletions', 'Insertions')),
    Sample = factor(Sample, levels = plt.order)
  ) %>%
  ggplot(aes(x=Sample, y=Fold_2, color=Treatment)) +
  geom_boxplot() +
  facet_wrap(~ Type, ncol = 1) +
  theme(
    legend.position='bottom',
    legend.text=element_text(size=rel(1.5)),
    legend.title=element_text(size=rel(2.25)),
    axis.text.x=element_text(angle=305, hjust=0.15, vjust=0.90, size=rel(0.85)),
    axis.title.x=element_blank()
  ) +
  guides(colour = guide_legend(override.aes = list(size=2))) +
  scale_color_manual(
    name = 'Treatment',
    values=c("#ca0020", "#0571b0")
  ) +
```

```
  labs(y='Log2-fold Error Rate Change')

#-------------------------------------------------------------------------------
# Panel 2
# Specific call outs for ErrASE/T7 Endo
pan2 <- enzPref.idm %>%
  filter(
    Type == 'M',
    Treatment == '2',
    Sample %in% c('ErrASE', 'T7 EndoI (Fuhrmann)')
  ) %>%
  mutate(Sym = Diff %>% recode(AC = 'A/T -> C/G', TG='A/T -> C/G',
                               AG = 'A/T -> G/C', TC='A/T -> G/C',
                               AT = 'A/T -> T/A', TA='A/T -> T/A',
                               CA = 'C/G -> A/T', GT='C/G -> A/T',
                               CG = 'C/G -> G/C', GC='C/G -> G/C',
                               CT = 'C/G -> T/A', GA='C/G -> T/A')
  ) %>%
  ggplot(aes(x=Sym, y=Fold_2, color=Class)) +
  geom_boxplot(
    outlier.shape = NA,
    show.legend = FALSE
  ) +
  geom_point(
    position=position_jitter(),
    size = 0.25,
    alpha = 0.8
  ) +
  facet_wrap(~ Sample, ncol=1, scales='free_y') +
  theme(
    legend.position = 'bottom',
    axis.title.x=element_blank(),
    axis.text.x = element_text(angle = 315, vjust=0.5)
  ) +
  guides(colour = guide_legend(override.aes = list(size=5))) +
  scale_color_manual(values = c('#7b3294', '#008837')) +
  labs(y='Log2-fold Error Rate Change')

grid.arrange(
  LabelMaker(pan1, 'A)'),
  LabelMaker(pan2, 'B)'),
  ncol = 2,
  widths = c(0.67, 0.33)
)
```




```
# table version of plot
enzPref %>%
  filter(!Sample %in% c('Doped Oligo', 'Standard Oligo', 'Standard Oligo ErrASE')) %>%
  mutate(
    Type = case_when(Type == 'D' | Type == 'P' ~ 'Deletions',
                     Type == 'I' | Type == 'S' ~ 'Insertions',
                     TRUE ~ 'Mismatches')
  ) %>%
  group_by(Sample, Treatment, Type) %>%
  summarise(
    Mean=mean(Fold),
    Median=median(Fold)
  ) %>%
  ungroup() %>%
  arrange(Sample, Treatment, Type)
```

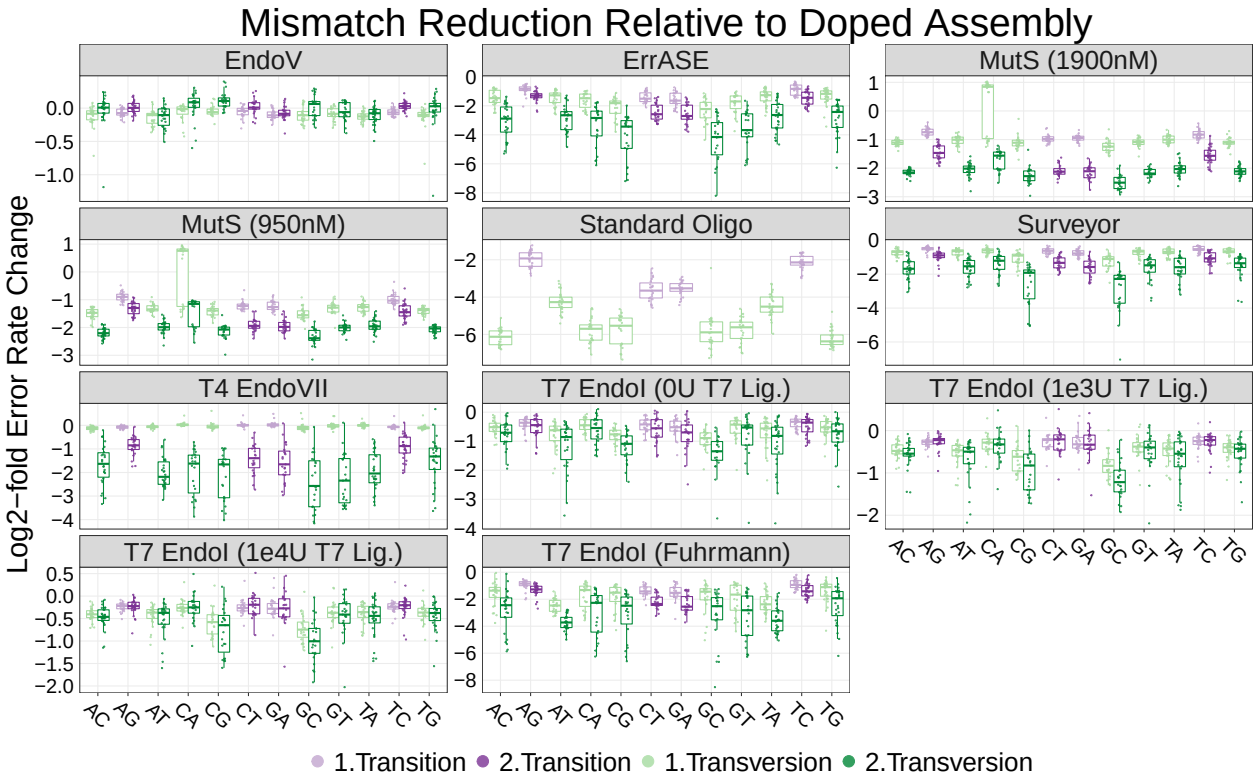
```
## Source: local data table [66 x 5]
##
## # tbl_dt [66 × 5]
##    Sample Treatment       Type       Mean    Median
##     <chr>     <chr>      <chr>      <dbl>     <dbl>
## 1   EndoV         1  Deletions   5.565447  4.152186
## 2   EndoV         1 Insertions  58.663793  5.906338
## 3   EndoV         1 Mismatches   1.064883  1.064428
## 4   EndoV         2  Deletions   5.845421  4.404096
## 5   EndoV         2 Insertions  38.613960  7.666019
## 6   EndoV         2 Mismatches   1.010861  1.008456
## 7  ErrASE         1  Deletions  68.864735 36.052656
```

```
## 8  ErrASE         1 Insertions 114.955902 40.655895
## 9  ErrASE         1 Mismatches   2.880278  2.510900
## 10 ErrASE         2  Deletions 188.193013 78.790890
## # ... with 56 more rows
```

Figure 4 - Supplement

Sup - Mismatch Preferences

```
enzPref.idm %>%
  filter(
    Type == 'M',
    !Sample %in% c('Doped Oligo', 'Standard Oligo (ErrASE)')
  ) %>%
  ggplot(
    aes(x=Diff,
        y=Fold_2,
        color=interaction(Treatment, Class))
  ) +
  geom_boxplot(
    outlier.shape = NA,
    show.legend = FALSE
  ) +
  geom_point(
    position=position_jitterdodge(),
    size = 0.25,
    alpha = 0.8
  ) +
  facet_wrap(~ Sample, ncol=3, scales='free_y') +
  theme(
    legend.position = 'bottom',
    axis.title.x=element_blank(),
    axis.text.x = element_text(angle = 315, vjust=0.5)
  ) +
  guides(colour = guide_legend(override.aes = list(size=5))) +
  scale_color_manual(values = c('#c2a5cf','#7b3294', '#a6dba0','#008837')) +
  labs(
    title='Mismatch Reduction Relative to Doped Assembly',
    y='Log2-fold Error Rate Change'
  )
```



There appears to be some real differences here, let's do some testing.

```
# first run anova to make sure there are diffs in medians
enzPref.idm %>%
  filter(
    Type %in% c('D', 'I', 'M'),
    !Sample %in% c('Doped Oligo', 'Standard Oligo', 'Standard Oligo (ErrASE)')
  ) %>%
  mutate(Diff = factor(Diff)) %>%
  group_by(Sample, Treatment, Type) %>%
  do(tidy(kruskal.test(Fold_2 ~ Diff, data=.))) %>%
  ungroup() %>%
  select(Sample, Treatment, Type, statistic, p.value)
```

```
## Source: local data table [60 x 5]
##
## # tbl_dt [60 × 5]
##                      Sample Treatment  Type statistic    p.value
##                       <chr>     <chr> <chr>     <dbl>      <dbl>
## 1                     EndoV         1     D  4.130051 0.24775910
## 2                    ErrASE         1     D  7.111247 0.06843521
## 3            MutS (1900nM)          1     D  3.926210 0.26954313
## 4             MutS (950nM)          1     D  9.139608 0.02749117
## 5                 Surveyor          1     D  6.943758 0.07371217
## 6               T4 EndoVII          1     D  3.826506 0.28081807
## 7    T7 EndoI (0U T7 Lig.)          1     D  2.789177 0.42528460
## 8  T7 EndoI (1e3U T7 Lig.)          1     D  4.480190 0.21406410
## 9  T7 EndoI (1e4U T7 Lig.)          1     D  3.789947 0.28505758
## 10     T7 EndoI (Fuhrmann)          1     D  1.974915 0.57762954
```

```
## # ... with 50 more rows
```

```
# transition vs transversions
enzPref.idm %>%
  filter(
    Type == 'M',
    !Sample %in% c('Doped Oligo', 'Standard Oligo', 'Standard Oligo (ErrASE)')
  ) %>%
  group_by(Sample, Treatment) %>%
  do(tidy(wilcox.test(Fold_2 ~ Class, data=.))) %>%
  ungroup() %>%
  select(Sample, Treatment, statistic, p.value)
```

```
## Source: local data table [20 x 4]
##
## # tbl_dt [20 × 4]
##                       Sample Treatment statistic      p.value
##                        <chr>     <chr>     <dbl>        <dbl>
## 1                     EndoV         1     11650 1.986610e-02
## 2                    ErrASE         1     14421 4.343993e-10
## 3            MutS (1900nM)          1     15318 6.023112e-14
## 4             MutS (950nM)          1     15836 1.737675e-16
## 5                 Surveyor          1     14181 3.584885e-09
## 6               T4 EndoVII          1     11555 2.818224e-02
## 7    T7 EndoI (0U T7 Lig.)          1     13773 1.002606e-07
## 8  T7 EndoI (1e3U T7 Lig.)          1     15384 2.943936e-14
## 9  T7 EndoI (1e4U T7 Lig.)          1     15151 3.547354e-13
## 10     T7 EndoI (Fuhrmann)         1     15033 1.201505e-12
## 11                    EndoV         2     10348 6.236937e-01
## 12                   ErrASE         2     16455 8.023391e-20
## 13           MutS (1900nM)          2     14586 9.538181e-11
## 14            MutS (950nM)          2     15691 9.418908e-16
## 15                Surveyor          2     14587 9.449412e-11
## 16              T4 EndoVII          2     15043 1.084638e-12
## 17   T7 EndoI (0U T7 Lig.)          2     13539 5.856954e-07
## 18 T7 EndoI (1e3U T7 Lig.)          2     15360 3.823047e-14
## 19 T7 EndoI (1e4U T7 Lig.)          2     14644 5.527648e-11
## 20     T7 EndoI (Fuhrmann)         2     15721 6.661996e-16
```

```
# median values for paper
enzPref.idm %>%
  filter(Treatment == '2') %>%
  group_by(Sample, Type, Diff) %>%
  summarise(med = median(Fold)) %>%
  spread(Sample, med)
```

```
## Source: local data table [20 x 13]
## Groups:
##
## # grouped_dt [20 × 13]
##     Type  Diff     EndoV     ErrASE `MutS (1900nM)` `MutS (950nM)` `Standard Oligo (ErrASE)`  Survey
## *  <chr> <chr>     <dbl>      <dbl>           <dbl>          <dbl>                     <dbl>     <dbl
## 1      D     A 1.9407476  27.268578        5.612455       4.402694                 15.995512  6.6450
```

```
## 2      D     C 1.9375507  19.428743        5.216762       3.677288                30.117109  4.6586
## 3      D     G 3.1062003  64.141297        7.276764       5.240921                70.168749 13.3917
## 4      D     T 2.3074515  25.498670        6.174086       5.880878                46.239703 11.5293
## 5      I     A 8.2996398  77.855939       14.969087      21.842308                69.359554  8.7819
## 6      I     C 9.1831863  87.715424       16.608982      20.562173                73.058730 13.7681
## 7      I     G 7.0873865 108.114360       30.762161      21.748965               117.911242 12.8416
## 8      I     T 8.2498679  83.781344       27.466216      19.714083                75.965226  8.7528
## 9      M    AC 0.9949576   7.387999        4.444978       4.610612                56.765547  3.2042
## 10     M    AG 0.9975086   2.463517        2.767391       2.459252                 3.638588  1.8559
## 11     M    AT 1.0751901   6.289659        4.047035       3.971451                16.447679  2.8975
## 12     M    CA 0.9410410   7.160285        2.954518       2.219500                45.379807  2.3073
## 13     M    CG 0.9283197  10.818895        4.857978       4.210227                77.405262  3.7755
## 14     M    CT 0.9944130   5.989511        4.358451       3.839514                12.789122  2.5174
## 15     M    GA 1.0612757   6.510108        4.312037       3.941436                12.406390  2.9803
## 16     M    GC 0.9575938  17.784310        5.686389       5.229828                97.387928  4.8557
## 17     M    GT 1.0426257  12.786200        4.560414       3.995814                46.507634  2.7841
## 18     M    TA 1.0551532   6.222358        4.084956       3.885780                18.062219  3.0059
## 19     M    TC 0.9800512   2.731725        2.966423       2.729911                 4.165589  2.0923
## 20     M    TG 0.9808719   5.373149        4.325695       4.099834                56.282996  2.5589
```

```
# ErrASE cg/gc, ag/tc
errase.pref <- enzPref.idm %>%
  filter(
    Sample == 'ErrASE',
    Type == 'M'
  ) %>%
  mutate(
    GC = if_else(Diff == 'GC' | Diff == 'CG', 'GC', 'No'),
    AG = if_else(Diff == 'AG' | Diff == 'TC', 'AG', 'No')
  ) %>%
  select(Pos, Diff, Treatment, Fold, GC, AG) %>%
  gather(Test, Val, GC, AG)

# p.val invariant of fold vs fold_2
errase.pref %>%
  group_by(Treatment, Test) %>%
  do(tidy(wilcox.test(Fold ~ Val, data=.))) %>%
  ungroup() %>%
  select(Test, Treatment, statistic, p.value)
```

```
## # A tibble: 4 × 4
##    Test Treatment statistic      p.value
##   <chr>     <chr>     <dbl>        <dbl>
## 1    AG         1      1704 1.192852e-17
## 2    GC         1      9647 2.100637e-12
## 3    AG         2       779 3.113673e-24
## 4    GC         2      9489 1.630541e-11
```

```
errase.pref %>%
  group_by(Test, Treatment, Val) %>%
  summarise(med = median(Fold))
```

```
## Source: local data frame [8 x 4]
```

```
## Groups: Test, Treatment [?]
##
##    Test Treatment   Val       med
##   <chr>     <chr> <chr>     <dbl>
## 1    AG         1    AG  1.776546
## 2    AG         1    No  2.922308
## 3    AG         2    AG  2.600594
## 4    AG         2    No  7.147050
## 5    GC         1    GC  3.949500
## 6    GC         1    No  2.422313
## 7    GC         2    GC 15.203511
## 8    GC         2    No  5.401971
```

```
# T7 ta/at, cg/gc(ligase), ag/tc
t7.pref <- enzPref.idm %>%
  filter(
    str_detect(Sample, 'T7'),
    Type == 'M'
  ) %>%
  mutate(
    AT = if_else(Diff == 'AT' | Diff == 'TA', 'AT', 'No'),
    GC = if_else(Diff == 'GC' | Diff == 'CG', 'GC', 'No'),
    AG = if_else(Diff == 'AG' | Diff == 'TC', 'AG', 'No')
  ) %>%
  select(Sample, Pos, Diff, Treatment, Fold, AT, GC, AG) %>%
  gather(Test, Val, AT, GC, AG)

t7.pref %>%
  group_by(Sample, Treatment, Test) %>%
  do(tidy(wilcox.test(Fold ~ Val, data=.))) %>%
  ungroup() %>%
  select(Sample, Test, Treatment, statistic, p.value)
```

```
## # A tibble: 24 × 5
##                     Sample  Test Treatment statistic      p.value
##                      <chr> <chr>     <chr>     <dbl>        <dbl>
## 1    T7 EndoI (0U T7 Lig.)    AG         1      3750 5.466407e-07
## 2    T7 EndoI (0U T7 Lig.)    AT         1      8189 7.381498e-03
## 3    T7 EndoI (0U T7 Lig.)    GC         1      9446 2.807120e-11
## 4    T7 EndoI (0U T7 Lig.)    AG         2      3869 1.562894e-06
## 5    T7 EndoI (0U T7 Lig.)    AT         2      8248 5.413589e-03
## 6    T7 EndoI (0U T7 Lig.)    GC         2      9150 9.989281e-10
## 7  T7 EndoI (1e3U T7 Lig.)    AG         1      3424 2.490561e-08
## 8  T7 EndoI (1e3U T7 Lig.)    AT         1      8157 8.698338e-03
## 9  T7 EndoI (1e3U T7 Lig.)    GC         1      9664 1.676603e-12
## 10 T7 EndoI (1e3U T7 Lig.)    AG         2      3532 7.171005e-08
## # ... with 14 more rows
```

```
t7.pref %>%
  group_by(Sample, Test, Treatment, Val) %>%
  summarise(med = median(Fold))
```

```
## Source: local data frame [48 x 5]
```

```
## Groups: Sample, Test, Treatment [?]
##
##                   Sample  Test Treatment   Val      med
##                    <chr> <chr>     <chr> <chr>    <dbl>
## 1  T7 EndoI (0U T7 Lig.)    AG         1    AG 1.267042
## 2  T7 EndoI (0U T7 Lig.)    AG         1    No 1.486359
## 3  T7 EndoI (0U T7 Lig.)    AG         2    AG 1.345917
## 4  T7 EndoI (0U T7 Lig.)    AG         2    No 1.716004
## 5  T7 EndoI (0U T7 Lig.)    AT         1    AT 1.522887
## 6  T7 EndoI (0U T7 Lig.)    AT         1    No 1.419058
## 7  T7 EndoI (0U T7 Lig.)    AT         2    AT 1.789891
## 8  T7 EndoI (0U T7 Lig.)    AT         2    No 1.605018
## 9  T7 EndoI (0U T7 Lig.)    GC         1    GC 1.752622
## 10 T7 EndoI (0U T7 Lig.)    GC         1    No 1.396459
## # ... with 38 more rows
```

```
# muts ag/tc cg/gc
muts.pref <- enzPref.idm %>%
  filter(
    str_detect(Sample, 'MutS'),
    Type == 'M'
  ) %>%
  mutate(
    GC = if_else(Diff == 'GC' | Diff == 'CG', 'GC', 'No'),
    AG = if_else(Diff == 'AG' | Diff == 'TC', 'AG', 'No')
  ) %>%
  select(Sample, Pos, Diff, Treatment, Fold, GC, AG) %>%
  gather(Test, Val, GC, AG)

muts.pref %>%
  group_by(Sample, Treatment, Test) %>%
  do(tidy(wilcox.test(Fold ~ Val, data=.))) %>%
  ungroup() %>%
  select(Sample, Test, Treatment, statistic, p.value)
```

```
## # A tibble: 8 × 5
##          Sample  Test Treatment statistic      p.value
##           <chr> <chr>     <chr>     <dbl>        <dbl>
## 1 MutS (1900nM)    AG         1      2150 7.196440e-15
## 2 MutS (1900nM)    GC         1      9248 3.162290e-10
## 3 MutS (1900nM)    AG         2       935 4.796316e-23
## 4 MutS (1900nM)    GC         2      9825 1.888872e-13
## 5  MutS (950nM)    AG         1      1533 8.763013e-19
## 6  MutS (950nM)    GC         1      8941 1.043401e-08
## 7  MutS (950nM)    AG         2      1005 1.598121e-22
## 8  MutS (950nM)    GC         2      9371 7.134109e-11
```

```
muts.pref %>%
  group_by(Sample, Test, Treatment, Val) %>%
  summarise(med = median(Fold))
```

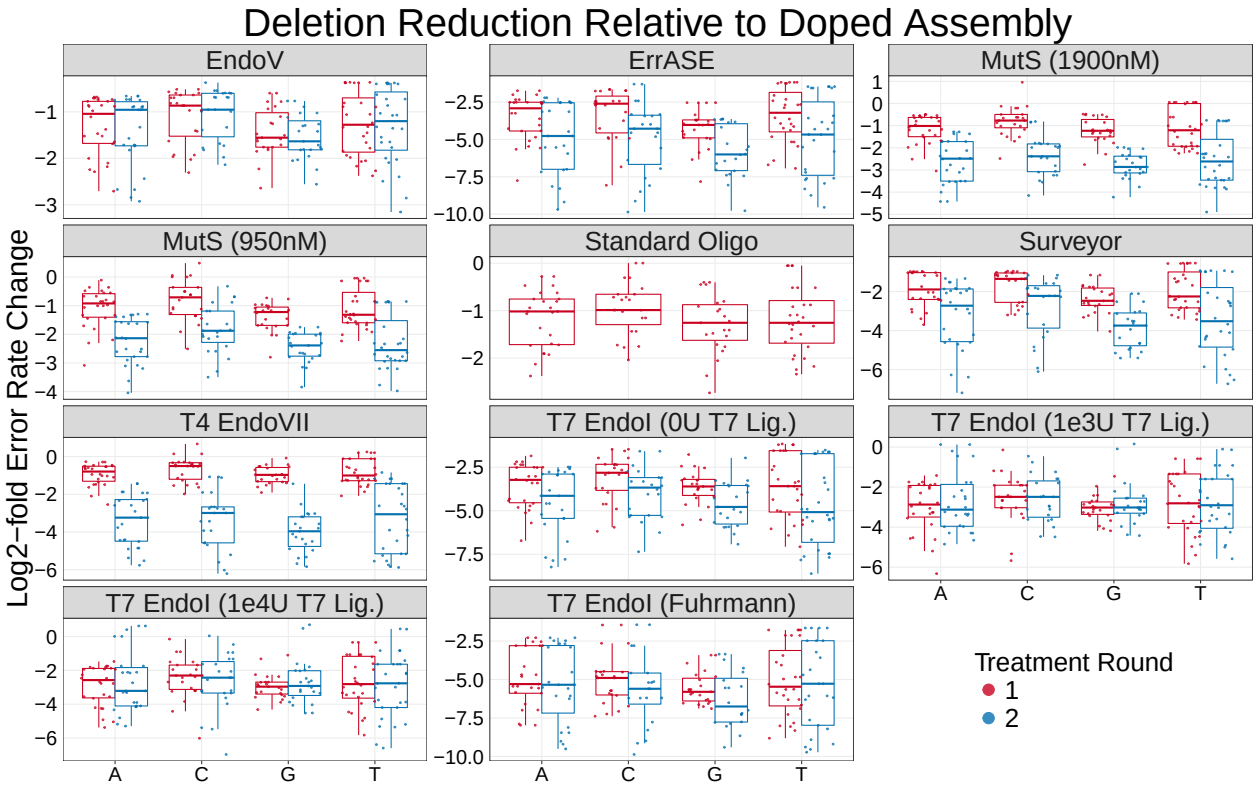
```
## Source: local data frame [16 x 5]
## Groups: Sample, Test, Treatment [?]
```

```
##
##           Sample  Test Treatment   Val      med
##            <chr> <chr>     <chr> <chr>    <dbl>
## 1  MutS (1900nM)    AG         1    AG 1.744946
## 2  MutS (1900nM)    AG         1    No 2.074040
## 3  MutS (1900nM)    AG         2    AG 2.805817
## 4  MutS (1900nM)    AG         2    No 4.357599
## 5  MutS (1900nM)    GC         1    GC 2.296331
## 6  MutS (1900nM)    GC         1    No 1.962474
## 7  MutS (1900nM)    GC         2    GC 5.130657
## 8  MutS (1900nM)    GC         2    No 4.092990
## 9   MutS (950nM)    AG         1    AG 1.958103
## 10  MutS (950nM)    AG         1    No 2.527475
## 11  MutS (950nM)    AG         2    AG 2.610612
## 12  MutS (950nM)    AG         2    No 4.070371
## 13  MutS (950nM)    GC         1    GC 2.758041
## 14  MutS (950nM)    GC         1    No 2.373760
## 15  MutS (950nM)    GC         2    GC 4.656740
## 16  MutS (950nM)    GC         2    No 3.798405
```

Sup - Deletion Preferences

Enzyme specificties for single base deletions

```
enzPref.idm %>%
  filter(
    Type == 'D',
    !Sample %in% c('Doped Oligo', 'Standard Oligo (ErrASE)')
  ) %>%  ggplot(aes(x=Diff, y=Fold_2, color=Treatment)) +
  geom_boxplot(
    outlier.shape = NA,
    show.legend = FALSE
  ) +
  geom_point(
    position=position_jitterdodge(),
    size = 0.5,
    alpha = 0.8
  ) +
  facet_wrap(~ Sample, ncol=3, scales='free_y') +
  theme(
    axis.title.x=element_blank(),
    legend.title=element_text(size=rel(2)),
    legend.position=c(0.85, 0.10)
    ) +
  guides(colour = guide_legend(override.aes = list(size=5))) +
  scale_color_manual(
    name = 'Treatment Round',
    values = c('#ca0020', '#0571b0')
  ) +
  labs(
    title='Deletion Reduction Relative to Doped Assembly',
       y='Log2-fold Error Rate Change'
  )
```

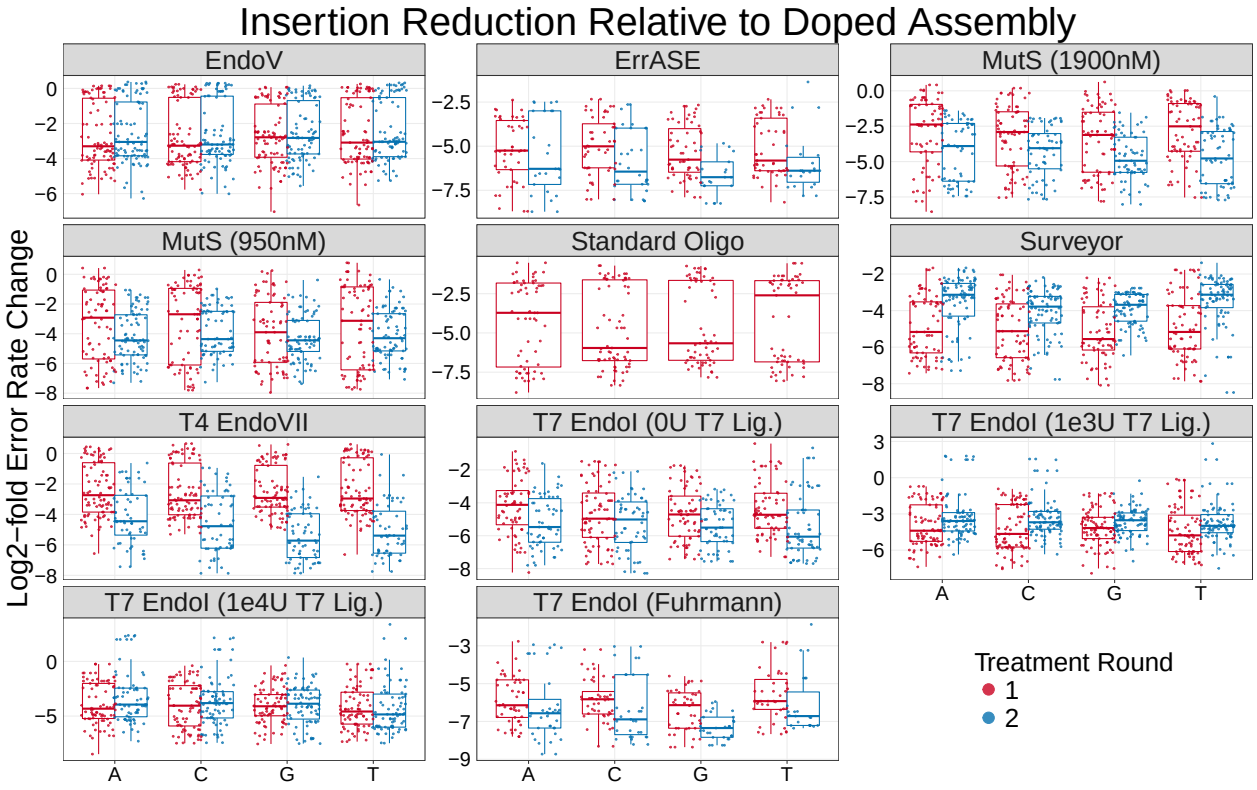


Sup - Insertion Preferences

Enzyme specificties for single base insertions

```
enzPref.idm %>%
  filter(
    Type == 'I',
    !Sample %in% c('Doped Oligo', 'Standard Oligo (ErrASE)')
  ) %>%  ggplot(aes(x=Diff, y=Fold_2, color=Treatment)) +
  geom_boxplot(
    outlier.shape = NA,
    show.legend = FALSE
  ) +
  geom_point(
    position=position_jitterdodge(),
    size = 0.5,
    alpha = 0.8
  ) +
  facet_wrap(~ Sample, ncol=3, scales='free_y') +
  theme(
    axis.title.x=element_blank(),
    legend.title=element_text(size=rel(2)),
    legend.position=c(0.85, 0.10)
    ) +
  guides(colour = guide_legend(override.aes = list(size=5))) +
  scale_color_manual(
    name = 'Treatment Round',
    values = c('#ca0020', '#0571b0')
```

```
  ) +
  labs(
    title='Insertion Reduction Relative to Doped Assembly',
       y='Log2-fold Error Rate Change'
    )
```



Extra Supplement

Doped Oligo

Non-Doped w/ ErrASE

Can we figure out what the noise floor is for our method? Is there a difference between the standard oligo and its error corrected counterpart?

```
allSamps %>%
  filter(Sample %in% c('1_nonDoped', '1_ErrASE-nonDoped', '2_ErrASE-nonDoped')) %>%
  DistribUncert2() %>%
  count(Sample, Type, Pos, wt=FracCount) %>%
  ungroup() %>%
  left_join(readCounts, by='Sample') %>%
  mutate(
    Norm = n / Reads,
    Treatment = str_sub(Sample, 1, 1),
    Sample = str_sub(Sample, 3),
    Sample = Sample %>%
```

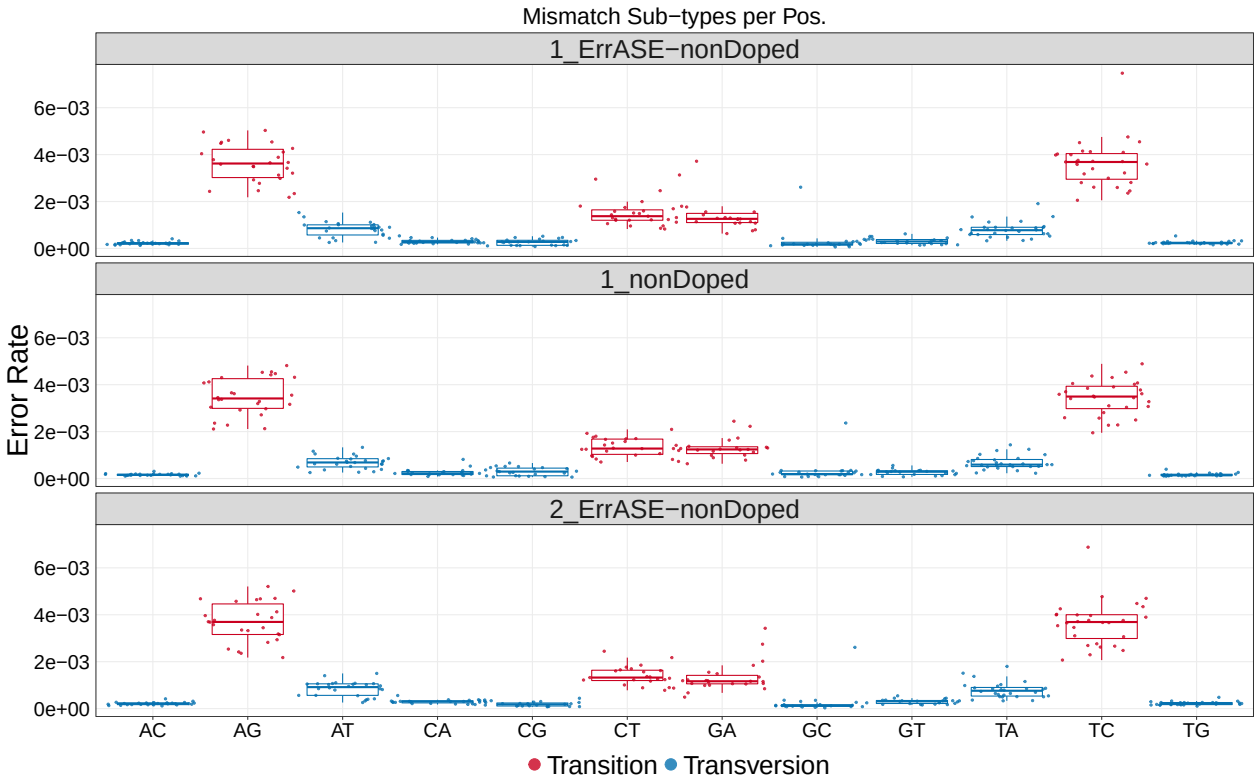
```
    recode(`ErrASE-nonDoped`='ErrASE',
           nonDoped='Standard Oligo'),
  Type = Type %>%
    factor(levels = c('M', 'I', 'D', 'P', 'S')) %>%
    recode(D = 'Single Base Deletions', I = 'Single Base Insertions',
           P = 'Multiple Base Deletions', S = 'Multiple Base Insertions',
           M = 'Mismatches')
) %T>%
{
  group_by(., Type) %>%
    do(with(.,
            tidy(pairwise.wilcox.test(Norm, interaction(Sample, Treatment))))) %>%
    print
} %>%
ggplot(aes(x=Sample, y=Norm, color=Treatment, group=Treatment)) +
facet_wrap(~ Type, ncol=2, scales='free_y') +
geom_point(
  size=0.75,
  alpha=0.8,
  position=position_jitterdodge()
) +
stat_summary(
  fun.y = median, fun.ymin = median, fun.ymax = median,
  geom = 'crossbar',
  width = 0.5,
  color='black',
  position=position_dodge(width=0.7)
) +
scale_y_log10() +
annotation_logticks(sides='l') +
theme(
  legend.title=element_text(size=rel(2)),
  legend.position=c(0.75, 0.15)
) +
guides(colour = guide_legend(override.aes = list(size=5))) +
scale_color_manual(
  name = 'Treatment',
  values=c("#ca0020", "#0571b0")
) +
labs(x = 'Position', y = 'Error Rate')
```

```
## Source: local data table [15 x 4]
## Groups: Type
##
## # grouped_dt [15 × 4]
##                         Type           group1           group2      p.value
##                       <fctr>           <fctr>            <chr>        <dbl>
## 1     Single Base Deletions Standard Oligo.1         ErrASE.1 2.806476e-28
## 2     Single Base Deletions         ErrASE.2         ErrASE.1 1.693221e-04
## 3     Single Base Deletions         ErrASE.2 Standard Oligo.1 1.782420e-29
## 4    Single Base Insertions Standard Oligo.1         ErrASE.1 4.223533e-20
## 5    Single Base Insertions         ErrASE.2         ErrASE.1 1.344410e-01
## 6    Single Base Insertions         ErrASE.2 Standard Oligo.1 8.154933e-04
```

```
## 7                Mismatches Standard Oligo.1         ErrASE.1 7.007785e-01
## 8                Mismatches         ErrASE.2         ErrASE.1 9.019551e-01
## 9                Mismatches         ErrASE.2 Standard Oligo.1 9.019551e-01
## 10  Multiple Base Deletions Standard Oligo.1         ErrASE.1 5.450814e-27
## 11  Multiple Base Deletions         ErrASE.2         ErrASE.1 2.052926e-11
## 12  Multiple Base Deletions         ErrASE.2 Standard Oligo.1 1.928125e-24
## 13 Multiple Base Insertions Standard Oligo.1         ErrASE.1 4.973757e-01
## 14 Multiple Base Insertions         ErrASE.2         ErrASE.1 3.150188e-02
## 15 Multiple Base Insertions         ErrASE.2 Standard Oligo.1 1.969505e-01
```


```
allSamps %>%
  filter(Sample %in% c('1_nonDoped', '1_ErrASE-nonDoped', '2_ErrASE-nonDoped')) %>%
  filter(Type == 'M') %>%
  count(Sample, Pos, Diff) %>%
  ungroup() %>%
  left_join(readCounts, by = 'Sample') %>%
  mutate(
    Norm = n / Reads,
    Class= Diff %>%
      recode(AT='Transversion', AG='Transition', AC='Transversion',
             TA='Transversion', TG='Transversion', TC='Transition',
             GA='Transition', GT='Transversion', GC='Transversion',
             CA='Transversion', CT='Transition', CG='Transversion')
  ) %>%
  ggplot(aes(x = Diff, y = Norm, color=Class)) +
  geom_boxplot(outlier.shape = NA, show.legend = FALSE) +
  geom_jitter(position=position_jitter(w = 0.5), size=0.75, alpha=0.8) +
  facet_wrap(~ Sample, ncol = 1) +
```

```
  labs(
    y = 'Error Rate',
    title = 'Mismatch Sub-types per Pos.'
  ) +
  theme(
    legend.position='bottom',
    legend.key.size=unit(0.75, "cm"),
    axis.title.x=element_blank(),
    plot.title = element_text(size=rel(1.75))
  ) +
  scale_y_continuous(labels = scientific_format()) +
  scale_color_manual(values=c("#ca0020", "#0571b0")) +
  guides(colour = guide_legend(override.aes = list(size=5)))
```



Doped Analysis

Here we will run the same sorts of analysis as Figure 2 on the Doped oligo.

```
positions_dope <- doped %>%
  DistribUncert2() %>%
  count(Pos, Type, wt=FracCount) %>%
  filter(Type != 'S') %>%
  ungroup() %>%
  mutate(
    Norm = n / subset(readCounts, Sample == '1_nonDoped')$Reads,
    Type = Type %>%
      factor(levels = c('M', 'I', 'D', 'P')) %>%
```

```
      recode(D = 'Single Base Deletions', I = 'Single Base Insertions',
             P = 'Multiple Base Deletions', M = 'Mismatches')
  ) %>%
  ggplot(aes(x=Pos, y=Norm)) +
  geom_point(size=3) +
  facet_wrap(~ Type, ncol=2) +
  stat_smooth(se=F) +
  labs(x = 'Position',
       y = 'Error Rate',
       title = 'Error Rate vs. Position for Error Sub-types') +
  scale_y_log10(labels = scientific_format()) +
  annotation_logticks(sides='l')

#------------------------------------------------------------------------------
# Panel 1b
# Percentage of Error Subtypes

sub_type_dope <- doped %>%
  DistribUncert2() %>%
  count(Type, wt=FracCount) %>%
  mutate(
    Norm = n / sum(n) * 100,
    Type = Type %>%
      factor(levels = c('M', 'D', 'P', 'I', 'S'))  %>%
      recode(M = 'MM', D = 'Del.', P = 'M. Del.', I = 'Ins.', S = 'M. Ins.')
  ) %T>%
  {arrange(., -Norm) %>% print()} %>%
  ggplot(aes(x=Type, y=Norm)) +
  geom_bar(stat='identity') +
  theme(plot.title = element_text(size=rel(2))) +
  labs(
    y = 'Percent',
    x = 'Error Type',
    title = 'Percentage of Error Sub-types'
  )
```

```
## Source: local data table [5 x 3]
##
## # tbl_dt [5 × 3]
##      Type       n      Norm
##    <fctr>   <dbl>     <dbl>
## 1      MM 1620841 90.867916
## 2    Del.   54805  3.072489
## 3 M. Del.   48431  2.715149
## 4    Ins.   33058  1.853304
## 5 M. Ins.   26598  1.491142
```

```
#------------------------------------------------------------------------------
# Panel c
# plot the distribution of total mismatches per position
mm_freq_dope <- doped %>%
  filter(Type == 'M') %>%
  mutate(
```

```
    To = str_sub(Diff, 2, 2),
    From = str_sub(Diff, 1, 1)
  ) %>%
  count(Pos, From) %>%
  ungroup() %>%
  mutate(Norm = n / subset(readCounts, Sample == '1_nonDoped')$Reads) %T>%
  { # pairwise wilcox test and print median values for paper
    group_by(., From) %>%
      summarise(med=median(Norm)) %>%
      arrange(-med) %>%
      print; # <- ; critical for . to be interpreted correctly
    with(., pairwise.wilcox.test(n, From)) %>% print
  } %>%
  ggplot(aes(x = From, y = Norm)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(position=position_jitter(w = 0.5), size=0.75, alpha=0.8) +
  # stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median, geom = 'crossbar', width = 0.5) +
  labs(y = 'Error Rate per Base',
       title = 'Mismatches per Pos.') +
  theme(
    axis.text.x = element_text(size=28),
    axis.title.x = element_blank(),
    plot.title = element_text(size=rel(2.25))
  ) +
  scale_y_continuous(labels = scientific_format())
```

```
## Source: local data table [4 x 2]
##
## # tbl_dt [4 × 2]
##    From        med
##   <chr>      <dbl>
## 1     C 0.03606105
## 2     T 0.03467046
## 3     G 0.03361262
## 4     A 0.02965034
##
##  Pairwise comparisons using Wilcoxon rank sum test
##
## data:  n and From
##
##   A     C     G
## C 0.049 -     -
## G 0.206 0.489 -
## T 0.409 0.489 0.918
##
## P value adjustment method: holm
```

```
#-------------------------------------------------------------------------------
# Panel d
# what bases are most likely mutated to
# We will normallize by the total count in each "from" group
mm_type_dope <- doped %>%
  filter(Type == 'M') %>%
```

```
  count(Pos, Diff) %>%
  ungroup() %>%
  mutate(
    Char=str_sub(Diff, 1, 1),
    Class= Diff %>%
      recode(AT='Transversion', AG='Transition', AC='Transversion',
             TA='Transversion', TG='Transversion', TC='Transition',
             GA='Transition', GT='Transversion', GC='Transversion',
             CA='Transversion', CT='Transition', CG='Transversion')

  ) %>%
  mutate(Norm = n / subset(readCounts, Sample == '1_nonDoped')$Reads) %T>%
  {# significance testing and printing for paper
    group_by(., Diff) %>%
      summarise(med=median(Norm)) %>%
      arrange(-med) %>%
      print; # <- ; critical for . to be interpreted correctly
      with(., pairwise.wilcox.test(Norm, Diff)) %>% print
  } %>%
  ggplot(aes(x = Diff, y = Norm, color=Class)) +
  geom_boxplot(outlier.shape = NA, show.legend = FALSE) +
  geom_jitter(position=position_jitter(w = 0.5), size=0.75, alpha=0.8) +
  # stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median, geom = 'crossbar', width = 0.5,
  labs(
    y = 'Error Rate per Base',
    title = 'Mismatch Sub-types per Pos.'
    ) +
  theme(
    legend.position='bottom',
    legend.key.size=unit(0.75, "cm"),
    axis.title.x=element_blank(),
    axis.text.x = element_text(angle = 315, vjust=0.5),
    plot.title = element_text(size=rel(1.75))
  ) +
  scale_y_continuous(labels = scientific_format()) +
  scale_color_manual(values = c('#7b3294', '#008837')) +
  guides(colour = guide_legend(override.aes = list(size=5)))
```

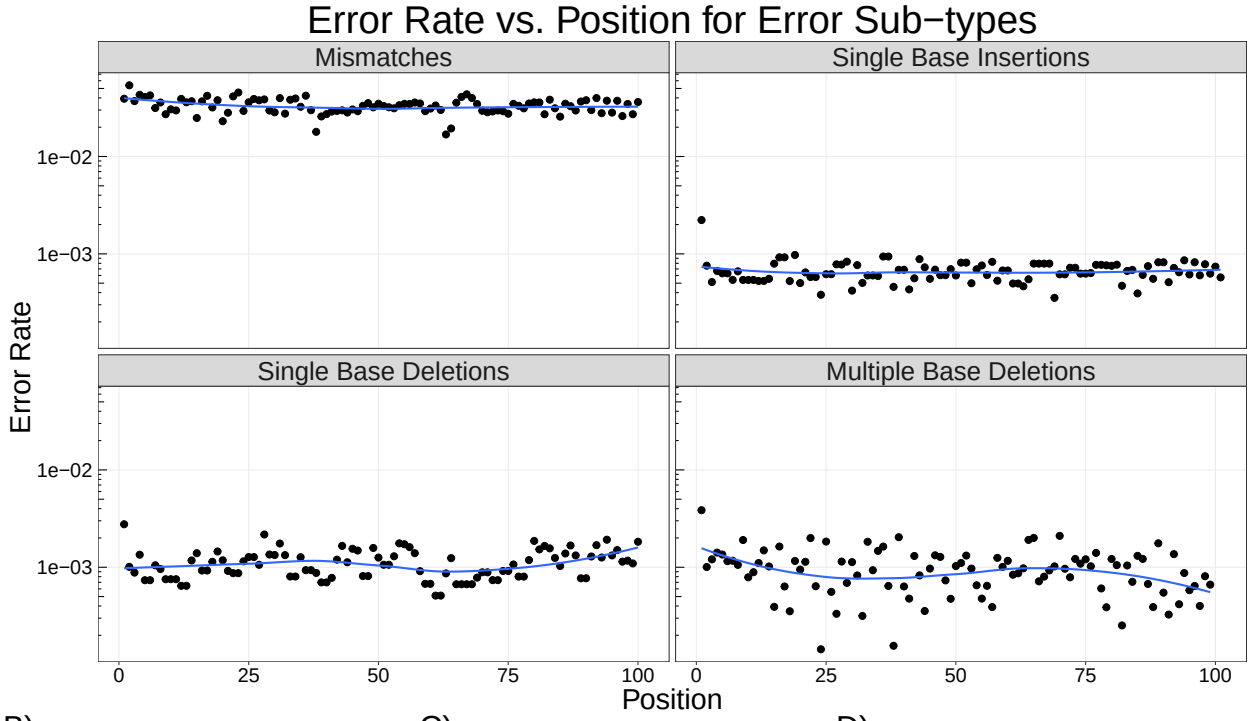
```
## Source: local data table [12 x 2]
##
## # tbl_dt [12 × 2]
##     Diff         med
##    <chr>       <dbl>
## 1     CT 0.014072356
## 2     GA 0.012556461
## 3     TC 0.012362352
## 4     TA 0.012078892
## 5     CG 0.011369215
## 6     CA 0.011200782
## 7     GT 0.011112458
## 8     AG 0.010847484
## 9     GC 0.010389429
## 10    AT 0.010046401
```

```
## 11    TG 0.009670508
## 12    AC 0.009366508
##
##  Pairwise comparisons using Wilcoxon rank sum test
##
## data:  Norm and Diff
##
##    AC      AG      AT      CA      CG      CT      GA      GC      GT      TA      TC
## AG 0.03251 -       -       -       -       -       -       -       -       -       -
## AT 1.00000 1.00000 -       -       -       -       -       -       -       -       -
## CA 0.02235 1.00000 1.00000 -       -       -       -       -       -       -       -
## CG 0.87035 1.00000 1.00000 1.00000 -       -       -       -       -       -       -
## CT 0.00022 0.50669 0.07019 0.81013 0.40479 -       -       -       -       -       -
## GA 0.00295 1.00000 1.00000 1.00000 1.00000 1.00000 -       -       -       -       -
## GC 1.00000 1.00000 1.00000 0.65909 1.00000 0.02070 0.02620 -       -       -       -
## GT 0.09648 1.00000 1.00000 1.00000 1.00000 0.81013 1.00000 1.00000 -       -       -
## TA 0.08809 1.00000 1.00000 1.00000 1.00000 1.00000 1.00000 1.00000 1.00000 -       -
## TC 1.7e-05 0.75909 0.20620 1.00000 1.00000 1.00000 1.00000 0.00730 1.00000 1.00000 -
## TG 1.00000 0.02293 1.00000 0.00730 1.00000 0.00025 0.00325 1.00000 0.07019 0.17116 3.4e-06
##
## P value adjustment method: holm
```

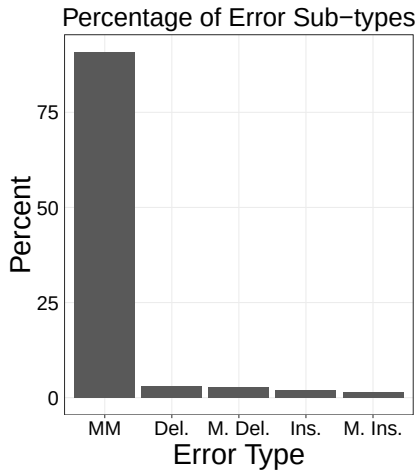
```
#-------------------------------------------------------------------------------
# plot everything!
grid.arrange(
  LabelMaker(positions_dope, 'A)'),
  arrangeGrob(
    LabelMaker(sub_type_dope, 'B)'),
    LabelMaker(mm_freq_dope, 'C)'),
    LabelMaker(mm_type_dope, 'D)'),
    nrow=1),
  ncol=1,
  heights = c(1, 0.67)
  )
```

```
## `geom_smooth()` using method = 'loess'
```

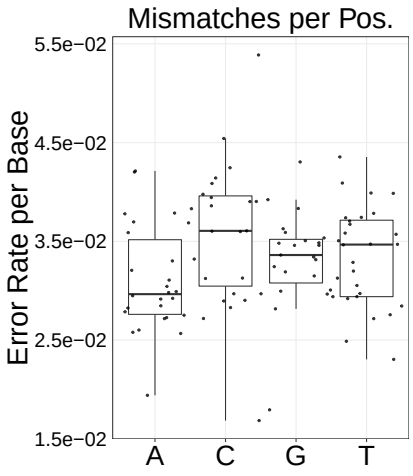
A)



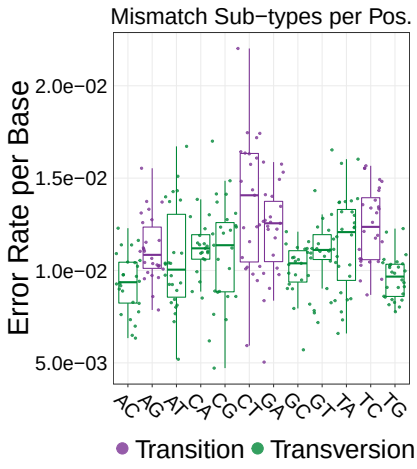
B)



C)



D)




```
#-------------------------------------------------------------------------------
# Values for text

# differences in medians in annealing regions and outside
doped %>%
  DistribUncert2() %>%
  count(Type, Pos, wt=FracCount) %>%
  mutate(Norm = n / subset(readCounts, Sample == '1_nonDoped')$Reads) %>%
  filter(Type == 'P') %>%
  mutate(Region = if_else(Pos >= 36 & Pos <=64, 'Anneal', 'No')) %T>%
  {wilcox.test(Norm ~ Region, data=.) %>% print} %>%
  group_by(Region) %>%
  summarise(med=median(Norm), IQR=IQR(Norm))
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Norm by Region
## W = 956.5, p-value = 0.6556
## alternative hypothesis: true location shift is not equal to 0

## Source: local data table [2 x 3]
##
## # tbl_dt [2 × 3]
##   Region          med          IQR
##    <chr>        <dbl>        <dbl>
## 1     No 0.0010126509 0.0005694878
## 2 Anneal 0.0009695157 0.0006059473
```

Here we will calculate the per-base error rate

```
doped %>%
  DistribUncert2() %>%
  count(Sample, Name, Type, wt=FracCount) %>%
  left_join(
    filter(charCounts, Sample == '1_DopedTemp'),
    by=c('Sample', 'Name')
  ) %>%
  group_by(Type) %>%
  summarise(m=sum(n/Len)/filter(readCounts, Sample == '1_DopedTemp')$Reads)
```

```
## Source: local data table [5 x 2]
##
## # tbl_dt [5 × 2]
##    Type            m
##   <chr>        <dbl>
## 1     M 0.0405901054
## 2     D 0.0015592068
## 3     I 0.0008250557
## 4     P 0.0015810213
## 5     S 0.0005879444
```

Doped vs Non-Doped Error Rates

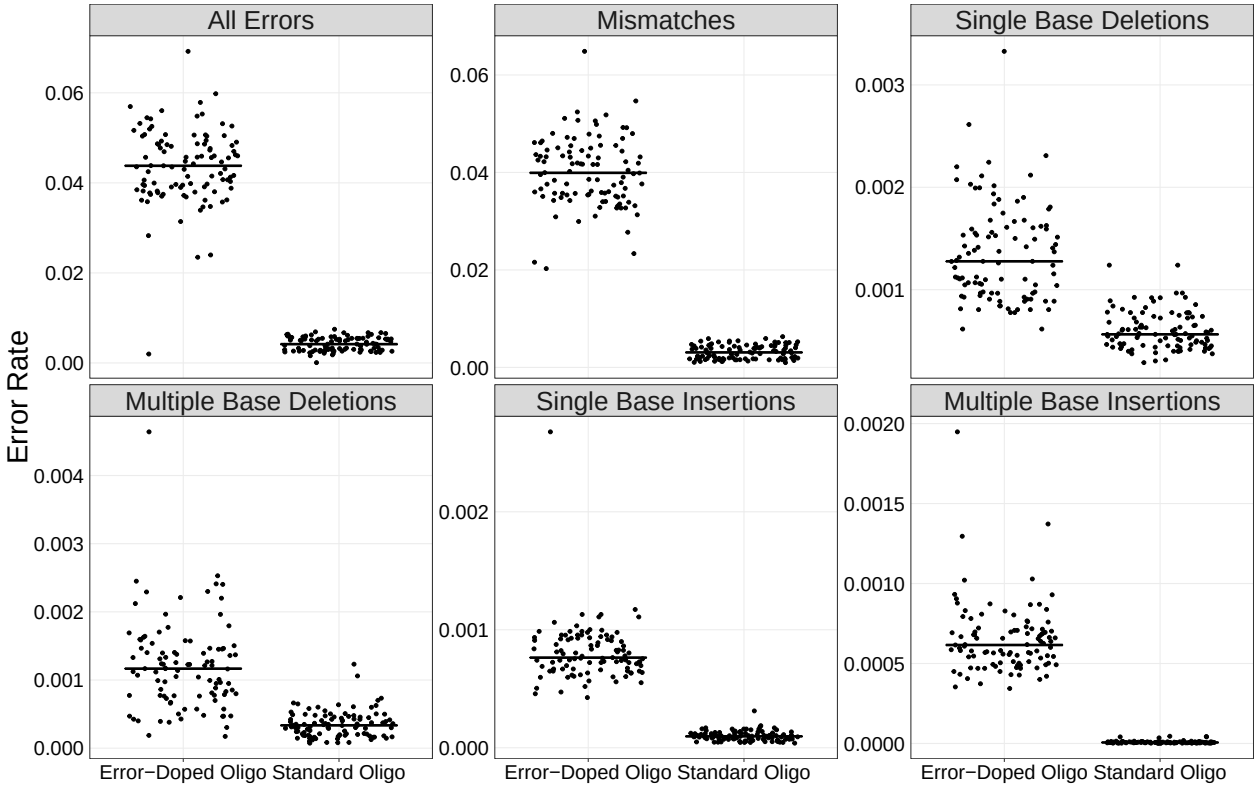
Here we compare the doped oligo to the non-doped. We see that all error rates are significantly higher in the doped sample.

```
allSamps %>%
  filter(Sample %in% c('1_DopedTemp', '1_nonDoped')) %>%
  DistribUncert2() %>%
  count(Sample, Type, Pos, wt=FracCount) %>%
  ungroup() %>%
  left_join(readCounts, by='Sample') %>%
  select(., -Errs) %>%
  # count all errors regardless of type
  bind_rows(.,
```

```
          count(., Sample, Reads, Pos, wt=n) %>% mutate(Type = 'A') %>% rename(n=nn)) %>%
  mutate(
    Norm = n / Reads,
    Sample = if_else(Sample == '1_DopedTemp',
                     'Error-Doped Oligo',
                     'Standard Oligo'),
    Type = Type %>%
      factor(levels = c('A', 'M', 'D', 'P', 'I', 'S')) %>%
      recode(D = 'Single Base Deletions', I = 'Single Base Insertions',
             P = 'Multiple Base Deletions', S = 'Multiple Base Insertions',
             M = 'Mismatches', A = 'All Errors')
  ) %T>%
  { # significance test and summarise for paper
    group_by(., Sample, Type) %>%
      summarise(med=median(Norm), mean=mean(Norm)) %>%
      arrange(Sample) %>%
      print();
    group_by(., Type) %>%
      summarise(p.val=wilcox.test(Norm ~ Sample, data=.)$p.value) %>%
        print()
    } %>%
  ggplot(aes(x=Sample, y=Norm)) +
  geom_jitter(position=position_jitter(w = 0.35)) +
  stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median, geom = 'crossbar', width = 0.75) +
  facet_wrap(~ Type, scales='free_y', ncol=3) +
  labs(y = 'Error Rate') +
  theme(axis.title.x=element_blank())
```

```
## Source: local data table [12 x 4]
## Groups: Sample
##
## # grouped_dt [12 × 4]
##               Sample                     Type          med         mean
##                <chr>                   <fctr>        <dbl>        <dbl>
## 1  Error-Doped Oligo    Single Base Deletions 1.276961e-03 1.354963e-03
## 2  Error-Doped Oligo   Single Base Insertions 7.639514e-04 8.092122e-04
## 3  Error-Doped Oligo               Mismatches 3.992449e-02 4.007261e-02
## 4  Error-Doped Oligo  Multiple Base Deletions 1.166942e-03 1.209471e-03
## 5  Error-Doped Oligo Multiple Base Insertions 6.156113e-04 6.510807e-04
## 6  Error-Doped Oligo               All Errors 4.379741e-02 4.366322e-02
## 7     Standard Oligo    Single Base Deletions 5.638391e-04 6.021062e-04
## 8     Standard Oligo   Single Base Insertions 9.654076e-05 9.959134e-05
## 9     Standard Oligo               Mismatches 3.079034e-03 3.189008e-03
## 10    Standard Oligo  Multiple Base Deletions 3.348116e-04 3.515968e-04
## 11    Standard Oligo Multiple Base Insertions 6.162176e-06 8.277247e-06
## 12    Standard Oligo               All Errors 4.175901e-03 4.206082e-03
## Source: local data table [6 x 2]
##
## # tbl_dt [6 × 2]
##                       Type        p.val
##                     <fctr>        <dbl>
## 1    Single Base Deletions 6.545164e-30
## 2   Single Base Insertions 1.194815e-34
```

```
## 3               Mismatches 2.561855e-34
## 4  Multiple Base Deletions 2.309269e-26
## 5 Multiple Base Insertions 8.611414e-35
## 6               All Errors 2.161343e-33
```



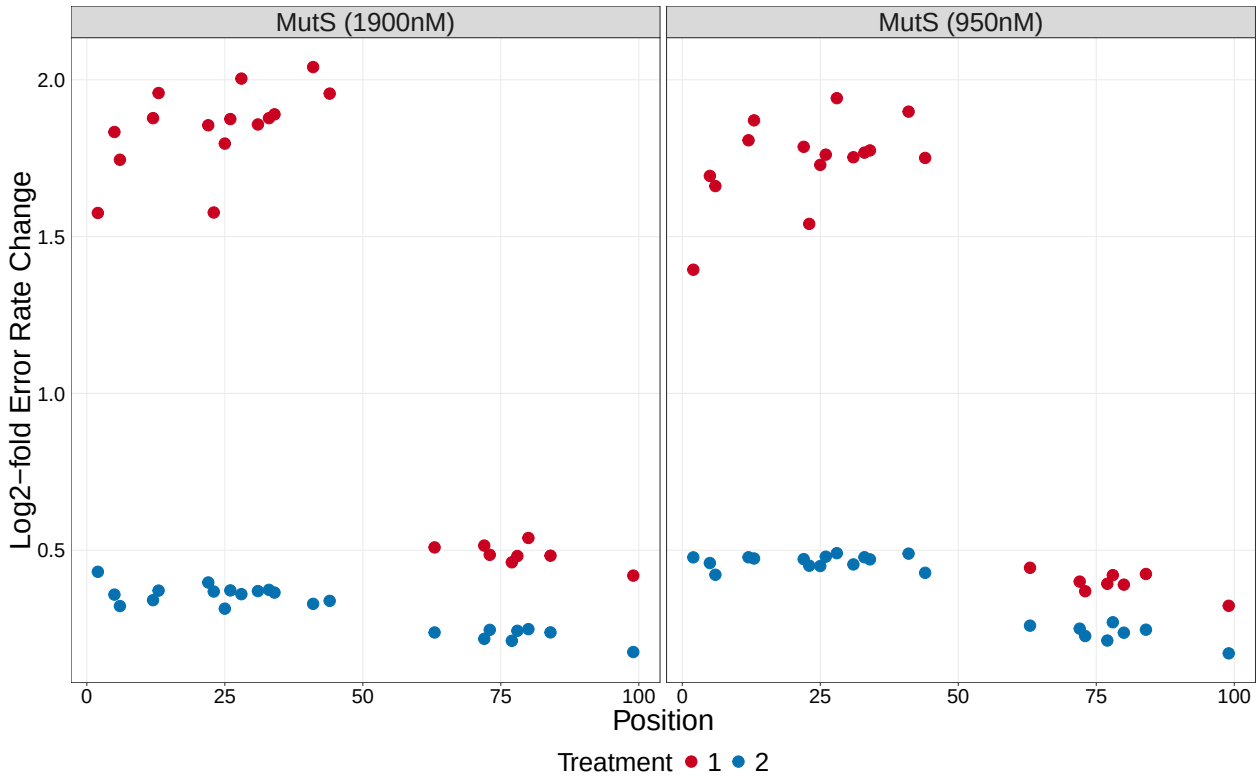
Misc

MutS Mismatch Preferences

There's some really strange bi-modal correction of CA mismatches with MutS across both samples. Could be guanine oxidization as suggested by Potapov 2017.

```
enzPref.idm %>%
  filter(
    str_detect(Sample, 'MutS'),
    Type == 'M',
    Diff == 'CA'
    ) %>%
  ggplot(aes(x=Pos, y=Rel_Norm, color=Treatment)) +
  geom_point(size=5) +
  facet_wrap(~ Sample, ncol=2) +
  guides(colour = guide_legend(override.aes = list(size=5))) +
  theme(
    legend.title = element_text(size=rel(2)),
    legend.position = 'bottom'
  ) +
```

```
  scale_color_manual(
    name = 'Treatment',
    values=c("#ca0020", "#0571b0")
  ) +
  labs(x = 'Position', y = 'Log2-fold Error Rate Change')
```



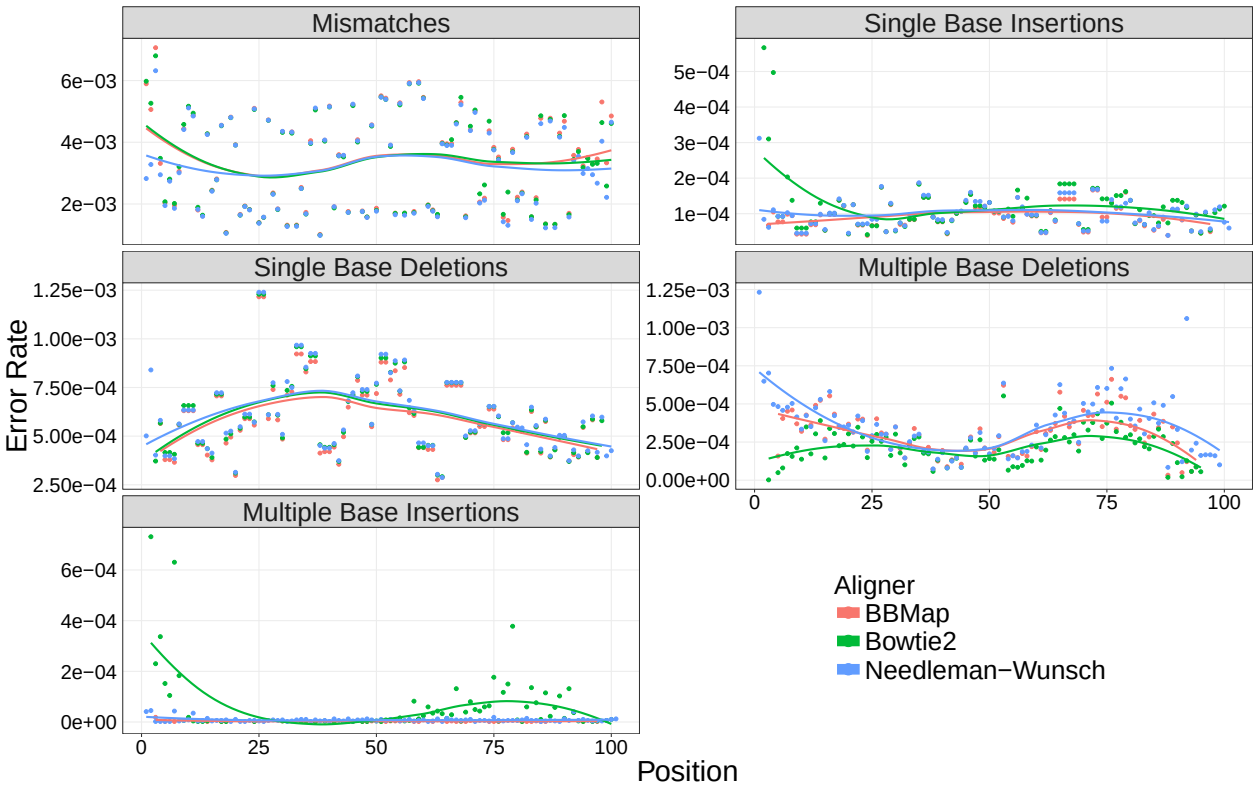
Aligner Comparision

Here we will compare BBMap, Bowtie2, and our NW aligner

```
bind_rows(
  list(bbmap=fread('./pipeline/1_nonDoped.bbmap.csv', header=T),
       bowtie=fread('./pipeline/1_nonDoped.bowtie.csv', header=T),
       nw=select(nonDoped, -Sample)),
  .id = 'Aligner'
) %>%
  DistribUncert2() %>%
  count(Aligner, Pos, Type, wt=FracCount) %>%
  ungroup() %>%
  mutate(
    Norm=n / subset(readCounts, Sample == '1_nonDoped')$Reads,
    Type = Type %>%
      factor(levels = c('M', 'I', 'D', 'P', 'S')) %>%
      recode(D = 'Single Base Deletions', I = 'Single Base Insertions',
             P = 'Multiple Base Deletions', S = 'Multiple Base Insertions',
             M = 'Mismatches')) %>%
  ggplot(aes(x=Pos, y=Norm, color=Aligner)) +
```

```
  facet_wrap(~ Type, ncol=2, scales='free_y') +
  geom_point() +
  stat_smooth(se=F) +
  theme(
    legend.title=element_text(size=rel(2)),
    legend.position=c(0.75, 0.15)
  ) +
  guides(colour = guide_legend(override.aes = list(size=5))) +
  scale_color_discrete(name = 'Aligner',
                       labels = c('BBMap', 'Bowtie2', 'Needleman-Wunsch')) +
  labs(x = 'Position', y = 'Error Rate') +
  scale_y_continuous(labels = scientific_format())
```

```
## `geom_smooth()` using method = 'loess'
```



Classic Table

```
single <- allSamps %>%
    filter(!Type %in% c('S', 'P')) %>%
      DistribUncert2() %>%
      group_by(Sample, Type, Diff) %>%
      summarise(n=sum(FracCount)) %>%
      ungroup()

# just count the number of multiple counts
```

```
multiple <- allSamps %>%
  filter(Type %in% c('S', 'P')) %>%
  group_by(Sample, Type) %>%
  summarise(n=n(), Diff='N/A') %>%
  ungroup()

# grab transitions/transversions
trans <- single %>%
  filter(Type == 'M') %>%
  mutate(
    Type= Diff %>%
      recode(AT='Transversion', AG='Transition', AC='Transversion',
             TA='Transversion', TG='Transversion', TC='Transition',
             GA='Transition', GT='Transversion', GC='Transversion',
             CA='Transversion', CT='Transition', CG='Transversion')
  ) %>%
  group_by(Sample, Type) %>%
  summarise(Diff='N/A', n=sum(n)) %>%
  ungroup()

bind_rows(single, multiple, trans) %>%
  mutate(
    Type = Type %>%
      recode(D = 'Single Base Deletions', I = 'Single Base Insertions',
             P = 'Multiple Base Deletions', S = 'Multiple Base Insertions',
             M = 'Mismatches')
  ) %>%
  spread(Sample, n) %>%
  # manual selection maddness to aid downstream processing
  select(
    Type, Diff,
    `1_nonDoped`, matches('ErrASE-'), `1_DopedTemp`,
    matches('_ErrASE'), contains("1900"), contains("950"),
    contains("Survey"), contains("Furhmann"), `1_T7EndoI`,
    `2_T7EndoI`, contains("e3T7"), contains("e4T7"),
    contains("T4"), contains("EndoV")
  ) %>%
  write.csv('Table_1.csv', quote=FALSE, row.names=FALSE)
```

Review

Figure 6

Error Rates

```
review.errRate <- review %>%
  DistribUncert2() %>%
  count(Sample, Name, wt=FracCount) %>%
  left_join(charCounts %>% filter(Sample %in% review.samples), ., by = c('Sample', 'Name')) %>%
  replace_na(list(n=0)) %>%
```

```
  group_by(Sample) %>%
  summarise(
    errRate = mean(n * (1000/Len)),
    sem = sd(n*(1000/Len)) / sqrt(n())
  ) %>%
  left_join(readCounts, by='Sample') %>%
  mutate(
    PercentPerf = (Reads - Errs) / Reads * 100,
    Construct = str_extract(Sample, "C([023])"),
    Polymerase = str_extract(Sample, "-(.*)-") %>% str_replace_all("-", ""),
    Repeat = str_extract(Sample, "\\d$")
  ) %>%
  group_by(Polymerase, Construct) %>%
  summarise(
    `Error / kb` = mean(errRate),
    `Percent Perfect` = mean(PercentPerf)
    # SEM = mean(sem),
    # PP_SEM = sd(PercentPerf) / n()
  ) %>%
  ungroup() %>%
  gather(Metric, Value, -Polymerase, -Construct)
```

```
review.errRate.plot <- review.errRate %>%
  # pretty print for fig
  mutate(
    Construct = if_else(Construct == 'C3', 'C1', 'C2'),
    Polymerase = if_else(Polymerase == 'Q5', 'Q5', 'KAPA2G Robust')
  ) %>%
  ggplot(aes(x=Construct, y=Value, fill = Polymerase)) +
  geom_bar(stat='identity', position = 'dodge') +
  facet_wrap(~ Metric, scales = 'free_y') +
  scale_fill_manual(
    name = 'Polymerase',
    values = c('#d8b365','#5ab4ac')
  ) +
  theme(
    legend.position = 'bottom',
    plot.title = element_text(size = rel(2)),
    axis.title.y = element_blank()
  ) +
  labs(title = 'Assembly Qualtiy vs. Polymerase')
```

Mismatch Sub-Type Preference

```
review.mm.subtype <- review %>%
  filter(Type == 'M') %>%
  count(Sample, Pos, Diff) %>%
  ungroup() %>%
  mutate(
    Char=str_sub(Diff, 1, 1),
    Construct = str_extract(Sample, "C([23])"),
```

```
    Polymerase = str_extract(Sample, "-(.*)-") %>% str_replace_all("-", ""),
    Repeat = str_extract(Sample, "\\d$"),
    Class=Diff %>%
      recode(AT='Transversion', AG='Transition', AC='Transversion',
             TA='Transversion', TG='Transversion', TC='Transition',
             GA='Transition', GT='Transversion', GC='Transversion',
             CA='Transversion', CT='Transition', CG='Transversion')
  ) %>%
  inner_join(review.counts, by = c('Construct', 'Char')) %>%
  inner_join(readCounts, by = 'Sample') %>%
  group_by(Construct, Polymerase, Diff, Pos) %>%
  summarise(
    Norm = mean(n / Count / Reads),
    Class = Class
  ) %>%
  ungroup()
```

```
review.subtype.plot <- review.mm.subtype %>%
  # pretty print for fig
  mutate(
    Construct = if_else(Construct == 'C3', 'C1', 'C2'),
    Polymerase = if_else(Polymerase == 'Q5', 'Q5', 'KAPA2G Robust')
  ) %>%
  ggplot(aes(x=Diff, y=Norm, color=Class)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(position = position_jitter(w=0.3), size = 0.5, alpha = 0.8) +
  facet_grid(Construct ~ Polymerase) +
  scale_y_log10() +
  annotation_logticks(sides = 'l') +
  scale_color_manual(values = c('#7b3294', '#008837')) +
  # guides(colour = guide_legend(override.aes = list(size=5))) +
  theme(
    legend.position = 'bottom',
    axis.text.x = element_text(size = rel(0.75))
  ) +
  labs(
    y = 'Error Rate',
    title = 'Mismatch Sub-types per Pos.'
  )

# test transitions vs transversions
review.mm.subtype %>%
  group_by(Construct, Polymerase) %>%
  do(tidy(wilcox.test(Norm ~ Class, data=.))) %>%
  select(-method)
```

```
## Source: local data table [4 x 5]
## Groups: Construct, Polymerase
##
## # grouped_dt [4 × 5]
##   Construct Polymerase statistic       p.value alternative
##       <chr>      <chr>     <dbl>         <dbl>      <fctr>
## 1        C2         Q5    352884 9.485010e-151   two.sided
```

```
## 2        C2        Taq    345400 1.388660e-119   two.sided
## 3        C3         Q5    322352  1.764183e-96   two.sided
## 4        C3        Taq    338816 3.918279e-110   two.sided
```

```
review.mm.subtype %>%
  group_by(Polymerase, Class, Construct) %>%
  summarise(Median = median(Norm)) %>%
  summarise(Mean = mean(Median))
```

```
## Source: local data table [4 x 3]
## Groups: Polymerase
##
## # grouped_dt [4 × 3]
##   Polymerase        Class         Mean
##        <chr>        <chr>        <dbl>
## 1         Q5   Transition 1.835474e-06
## 2         Q5 Transversion 4.687303e-07
## 3        Taq Transversion 6.394950e-06
## 4        Taq   Transition 5.319500e-05
```

```
# test Q5 C/G -> A/T vs transitions
review.mm.subtype %>%
  filter(Class == 'Transition' | Diff %in% c('CA', 'GT')) %>%
  mutate(Class = if_else(Class == 'Transversion', 'C/G -> A/T', Class)) %>%
  group_by(Construct, Polymerase, Class) %>%
  summarise(Med = median(Norm), N = n())
```

```
## Source: local data table [8 x 5]
## Groups: Construct, Polymerase
##
## # grouped_dt [8 × 5]
##   Construct Polymerase      Class          Med     N
##       <chr>      <chr>      <chr>        <dbl> <int>
## 1        C2         Q5 Transition 1.753501e-06   440
## 2        C2         Q5 C/G -> A/T 7.274858e-07   223
## 3        C2        Taq Transition 5.046917e-05   440
## 4        C2        Taq C/G -> A/T 5.825988e-06   224
## 5        C3         Q5 Transition 1.917447e-06   440
## 6        C3         Q5 C/G -> A/T 1.827342e-06   222
## 7        C3        Taq Transition 5.592084e-05   440
## 8        C3        Taq C/G -> A/T 5.930915e-06   222
```

```
review.mm.subtype %>%
  filter(Class == 'Transition' | Diff %in% c('CA', 'GT')) %>%
  mutate(Class = if_else(Class == 'Transversion', 'C/G -> A/T', Class)) %>%
  group_by(Construct, Polymerase) %>%
  do(tidy(wilcox.test(Norm ~ Class, data=.))) %>%
  select(-method)
```

```
## Source: local data table [4 x 5]
## Groups: Construct, Polymerase
##
```

```
## # grouped_dt [4 × 5]
##   Construct Polymerase statistic      p.value alternative
##       <chr>      <chr>     <dbl>        <dbl>      <fctr>
## 1        C2         Q5     11866 2.337299e-57   two.sided
## 2        C2        Taq      8756 2.365609e-67   two.sided
## 3        C3         Q5     46944 4.145362e-01   two.sided
## 4        C3        Taq      9712 1.184661e-63   two.sided
```

Overlap Effects

Are the decreases in multiple base deletions in the overlaps real?

```
overlaps <- c(seq(41, 60), seq(81, 100), seq(121, 140), seq(161, 180))
primers <- c(seq(1, 15), seq(206, 220))

overlap.data <- review %>%
  filter(Type == 'P') %>%
  count(Sample, Pos) %>%
  inner_join(readCounts, by = 'Sample') %>%
  mutate(
    Construct = str_extract(Sample, "C([23])"),
    Polymerase = str_extract(Sample, "-(.*)-") %>% str_replace_all("-", ""),
    Repeat = str_extract(Sample, "\\d$")
  ) %>%
  group_by(Construct, Polymerase, Pos) %>%
  summarise(Norm = mean(n / Reads)) %>%
  ungroup() %>%
  mutate(
    Overlap = case_when(Pos %in% primers ~ 'Primers',
                        Pos %in% overlaps ~ 'Overlaps',
                        TRUE ~ 'Other')
  )
```

```
overlap.plot <- overlap.data %>%
  # better naming for figure
  mutate(
    Construct = if_else(Construct == 'C3', 'C1', 'C2'),
    Polymerase = if_else(Polymerase == 'Q5', 'Q5', 'KAPA2G Robust')
  ) %>%
  ggplot(aes(x = Overlap, y = Norm, color=Polymerase)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(position = position_jitterdodge(), size = 0.5, alpha = 0.8) +
  facet_wrap(~ Construct) +
  scale_y_log10() +
  annotation_logticks(sides = 'l') +
  scale_color_manual(
    name = 'Polymerase',
    values = c('#d8b365','#5ab4ac')
  ) +
  theme(
    legend.position = 'bottom',
    plot.title = element_text(size = rel(1.75)),
    axis.text.x = element_text(size = rel(0.65))
```

```
  ) +
  labs(
    y = 'Error Rate',
    x = 'Region',
    title = 'Multiple Base Deletions per Base'
  )

# hypothesis test
overlap.data %>%
  group_by(Construct, Polymerase) %>%
  do(tidy(pairwise.wilcox.test(Norm, Overlap)))
```

```
## Source: local data table [12 x 5]
## Groups: Construct, Polymerase
##
## # grouped_dt [12 × 5]
##    Construct Polymerase   group1   group2      p.value
##        <chr>      <chr>   <fctr>    <chr>        <dbl>
## 1         C2         Q5 Overlaps    Other 7.893169e-10
## 2         C2         Q5  Primers    Other 7.893169e-10
## 3         C2         Q5  Primers Overlaps 6.614892e-06
## 4         C2        Taq Overlaps    Other 3.187023e-09
## 5         C2        Taq  Primers    Other 5.758465e-08
## 6         C2        Taq  Primers Overlaps 3.046185e-04
## 7         C3         Q5 Overlaps    Other 3.813500e-10
## 8         C3         Q5  Primers    Other 1.415076e-11
## 9         C3         Q5  Primers Overlaps 5.465786e-08
## 10        C3        Taq Overlaps    Other 1.405592e-08
## 11        C3        Taq  Primers    Other 4.471158e-11
## 12        C3        Taq  Primers Overlaps 6.755179e-06
```

```
overlap.data %>%
  filter(Overlap == 'Primers') %>%
  group_by(Construct) %>%
  do(tidy(wilcox.test(Norm ~ Polymerase, data=.)))
```

```
## Source: local data table [2 x 5]
## Groups: Construct
##
## # grouped_dt [2 × 5]
##   Construct statistic    p.value                                                method alternative
##       <chr>     <dbl>      <dbl>                                                 <fctr>      <fctr>
## 1        C2       225 0.13621424 Wilcoxon rank sum test with continuity correction   two.sided
## 2        C3       206 0.04045399 Wilcoxon rank sum test with continuity correction   two.sided
```

```
# median values
overlap.data %>%
  group_by(Construct, Polymerase, Overlap) %>%
  summarise(Median = median(Norm))
```

```
## Source: local data table [12 x 4]
## Groups: Construct, Polymerase
```

```
##
## # grouped_dt [12 × 4]
##    Construct Polymerase  Overlap       Median
##        <chr>      <chr>    <chr>        <dbl>
## 1         C2         Q5    Other 2.480612e-04
## 2         C2         Q5 Overlaps 1.215581e-04
## 3         C2         Q5  Primers 2.236011e-05
## 4         C2        Taq    Other 2.416478e-04
## 5         C2        Taq Overlaps 1.031826e-04
## 6         C2        Taq  Primers 4.277067e-05
## 7         C3         Q5    Other 2.965672e-04
## 8         C3         Q5 Overlaps 1.595198e-04
## 9         C3         Q5  Primers 2.071808e-05
## 10        C3        Taq    Other 3.154898e-04
## 11        C3        Taq  Primers 5.293481e-05
## 12        C3        Taq Overlaps 1.619796e-04
```

```
# diff between medians
overlap.data %>%
  group_by(Construct, Polymerase, Overlap) %>%
  summarise(Median = median(Norm)) %>%
  ungroup() %>%
  spread(Overlap, Median) %>%
  gather(Region, Value, Overlaps, Primers) %>%
  mutate(Rate = Other / Value) %>%
  group_by(Polymerase, Region) %>%
  summarise(Mean = mean(Rate), sd = sd(Rate))
```

```
## Source: local data frame [4 x 4]
## Groups: Polymerase [?]
##
##   Polymerase   Region      Mean        sd
##        <chr>    <chr>     <dbl>     <dbl>
## 1         Q5 Overlaps  1.949902 0.1283797
## 2         Q5  Primers 12.704167 2.2772350
## 3        Taq Overlaps  2.144828 0.2787629
## 4        Taq  Primers  5.804909 0.2192879
```

Error Rate Table

Here we calculate the SEM in a more explicit fashion than above. Note that the left_join followed by a NA -> 0 above only really serves as a hack to make the built in mean/sd functions calculate the appropriate number of samples.

```
review.table <- nonDoped %>%
  bind_rows(review) %>%
  DistribUncert2() %>%
  count(Sample, Type, Name, wt=FracCount) %>%
  inner_join(charCounts, by = c('Sample', 'Name')) %>%
  summarise(
    Sum = sum(n * (1000/Len)),
    SumSq = sum((n * (1000/Len))^2)
  ) %>%
```

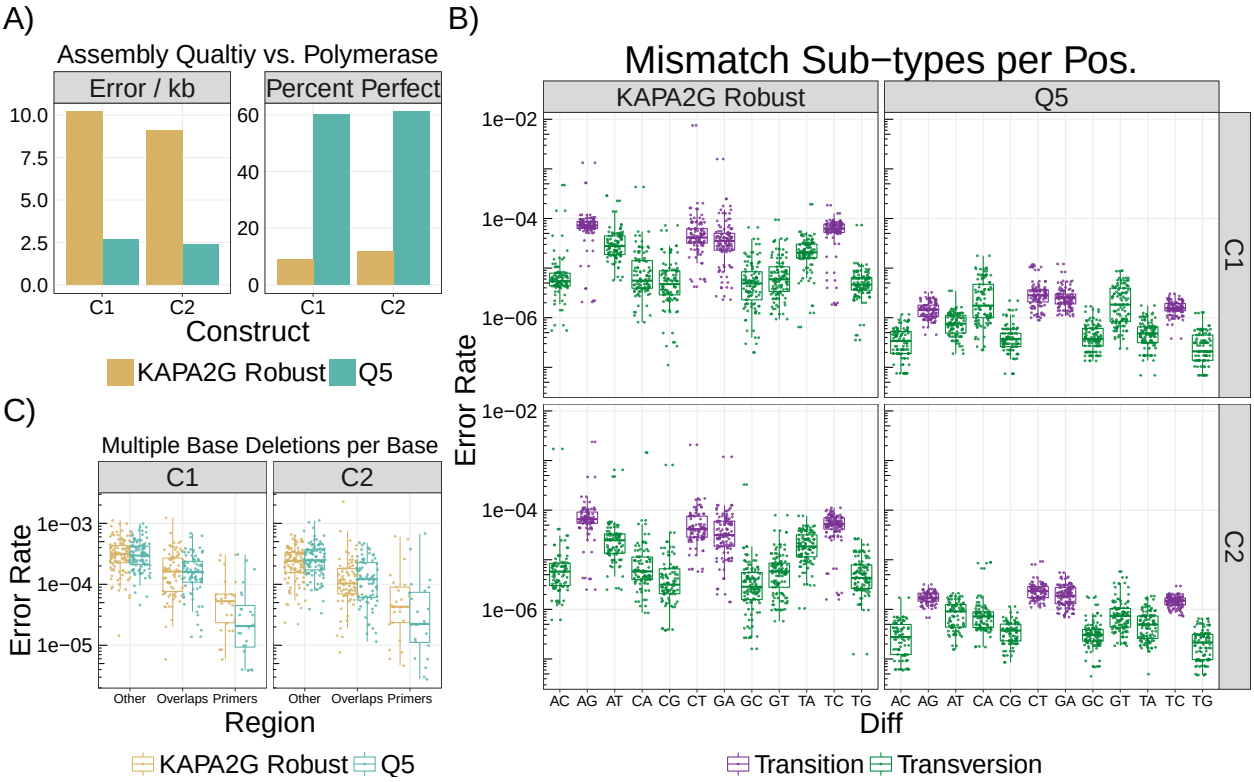
```
  inner_join(readCounts, by = 'Sample') %>%
  mutate(
    errRate = Sum / Reads,
    sd = sqrt((SumSq - (Sum^2 / Reads)) / (Reads - 1)),
    sem = sd / sqrt(Reads),
    Sample = recode(Sample, `1_nonDoped` = 'CO-Taq*-1'),
    Construct = str_extract(Sample, "C([023])"),
    Polymerase = str_extract(Sample, "-(.*)-") %>% str_replace_all("-", ""),
    Repeat = str_extract(Sample, "\\d$")
  ) %>%
  group_by(Polymerase, Type) %>%
  summarise(
    `Error Rate` = mean(errRate),
    SEM = mean(sem)
  ) %>%
  ungroup()

review.table %>%
  mutate(
    Value = paste(round(`Error Rate`, 4), "±", round(SEM, 4)),
    Type = Type %>%
      factor(levels = c('M', 'D', 'I', 'P', 'S')) %>%
      recode(D = 'Single Base Deletions', I = 'Single Base Insertions',
             P = 'Multiple Base Deletions', M = 'Mismatches',
             S = 'Multiple Base Insertions')
    ) %>%
  select(Polymerase, Type, Value) %>%
  spread(Polymerase, Value)
```

```
## Source: local data table [5 x 4]
##
## # tbl_dt [5 × 4]
##                        Type              Q5             Taq          `Taq*`
## *                     <fctr>           <chr>           <chr>           <chr>
## 1               Mismatches 0.2131 ± 0.0019 7.1388 ± 0.0121 3.1981 ± 0.0083
## 2    Single Base Deletions 2.0121 ± 0.0062  2.1891 ± 0.008 0.6121 ± 0.0038
## 3   Single Base Insertions 0.0747 ± 0.0011 0.0816 ± 0.0014 0.0991 ± 0.0016
## 4  Multiple Base Deletions  0.2326 ± 0.002 0.2342 ± 0.0029 0.3904 ± 0.0031
## 5 Multiple Base Insertions  0.0014 ± 2e-04  0.0083 ± 4e-04  0.0055 ± 7e-04
```

Actual Figure

```
grid.arrange(
  arrangeGrob(
    LabelMaker(review.errRate.plot, 'A)'),
    LabelMaker(overlap.plot, 'C)'),
    ncol = 1
  ),
  LabelMaker(review.subtype.plot, 'B)'),
  nrow = 1,
  widths = c(0.55, 1)
)
```



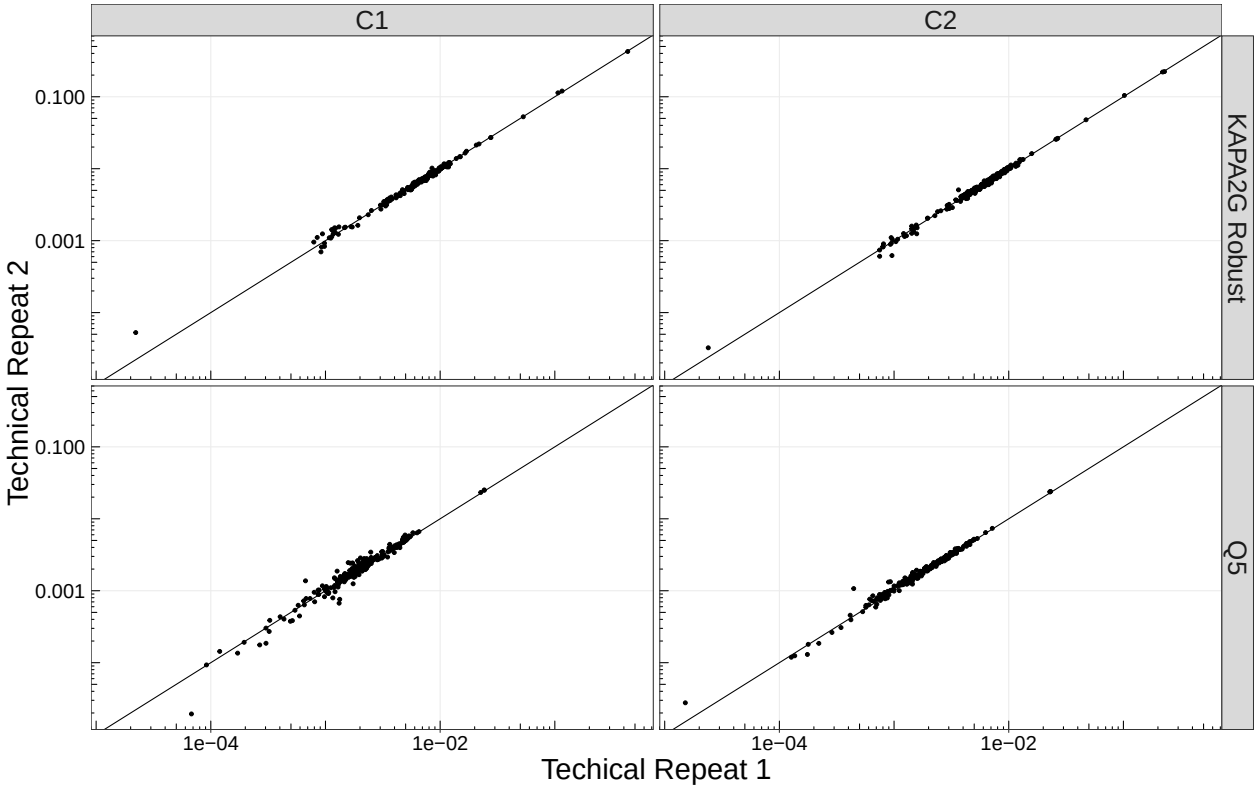
Suplement

Correlation between Repeats

```
review %>%
  DistribUncert2() %>%
  count(Sample, Pos, wt = FracCount) %>%
  ungroup() %>%
  inner_join(readCounts, by = 'Sample') %>%
  mutate(
    Norm = n / Reads,
    Construct = str_extract(Sample, "C([23])"),
    Polymerase = str_extract(Sample, "-(.*)-") %>% str_replace_all("-", ""),
    Repeat = str_extract(Sample, "\\d$")
  ) %>%
  select(-Sample, -Reads, -Errs, -n) %>%
  spread(Repeat, Norm) %T>%
  {
    group_by(., Construct, Polymerase) %>%
      summarise(Cor = cor(`1`, `2`)) %>%
      print()
  } %>%
  # pretty plot
  mutate(
    Construct = if_else(Construct == 'C3', 'C1', 'C2'),
    Polymerase = if_else(Polymerase == 'Q5', 'Q5', 'KAPA2G Robust')
  ) %>%
```

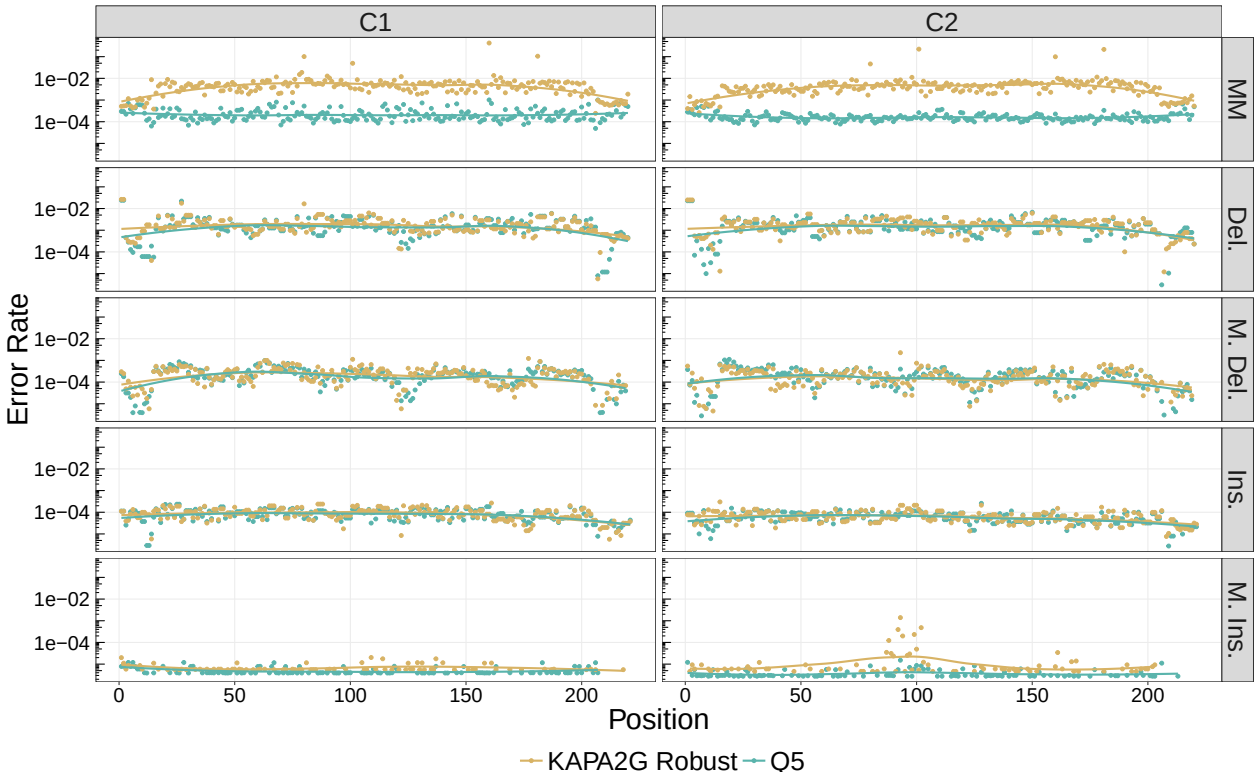
```
  ggplot(aes(x=`1`, y=`2`)) +
  geom_point() +
  geom_abline() +
  facet_grid(Polymerase ~ Construct) +
  scale_x_log10() +
  scale_y_log10() +
  annotation_logticks() +
  labs(
    x = 'Techical Repeat 1',
    y = 'Technical Repeat 2'
  )
```

```
## Source: local data table [4 x 3]
## Groups: Construct
##
## # grouped_dt [4 × 3]
##   Construct Polymerase       Cor
##       <chr>      <chr>     <dbl>
## 1        C2         Q5 0.9990096
## 2        C2        Taq 0.9998038
## 3        C3         Q5 0.9960850
## 4        C3        Taq 0.9997145
```



Classic Positional Graph

```
review %>%
  DistribUncert2() %>%
  count(Sample, Type, Pos, wt = FracCount) %>%
  ungroup() %>%
  mutate(
    Construct = str_extract(Sample, "C([23])"),
    Polymerase = str_extract(Sample, "-(.*)-") %>% str_replace_all("-", ""),
    Repeat = str_extract(Sample, "\\d$")
  ) %>%
  inner_join(readCounts, by = 'Sample') %>%
  group_by(Construct, Polymerase, Type, Pos) %>%
  summarise(Norm = mean(n / Reads)) %>%
  # better naming for figure
  ungroup() %>%
  mutate(
    Construct = if_else(Construct == 'C3', 'C1', 'C2'),
    Polymerase = if_else(Polymerase == 'Q5', 'Q5', 'KAPA2G Robust'),
    Type = Type %>%
      factor(levels = c('M', 'D', 'P', 'I', 'S'))  %>%
      recode(M = 'MM', D = 'Del.', P = 'M. Del.', I = 'Ins.', S = 'M. Ins.')
  ) %>%
  ggplot(aes(x=Pos, y=Norm, color=Polymerase)) +
  geom_point() +
  geom_smooth(se = F, method = 'loess') +
  facet_grid(Type ~ Construct) +
  scale_y_log10() +
  annotation_logticks(sides = 'l') +
  scale_color_manual(
    name = 'Polymerase',
    values = c('#d8b365','#5ab4ac')
  ) +
  theme(legend.position = 'bottom') +
  labs(
    x = 'Position',
    y = 'Error Rate'
  )
```

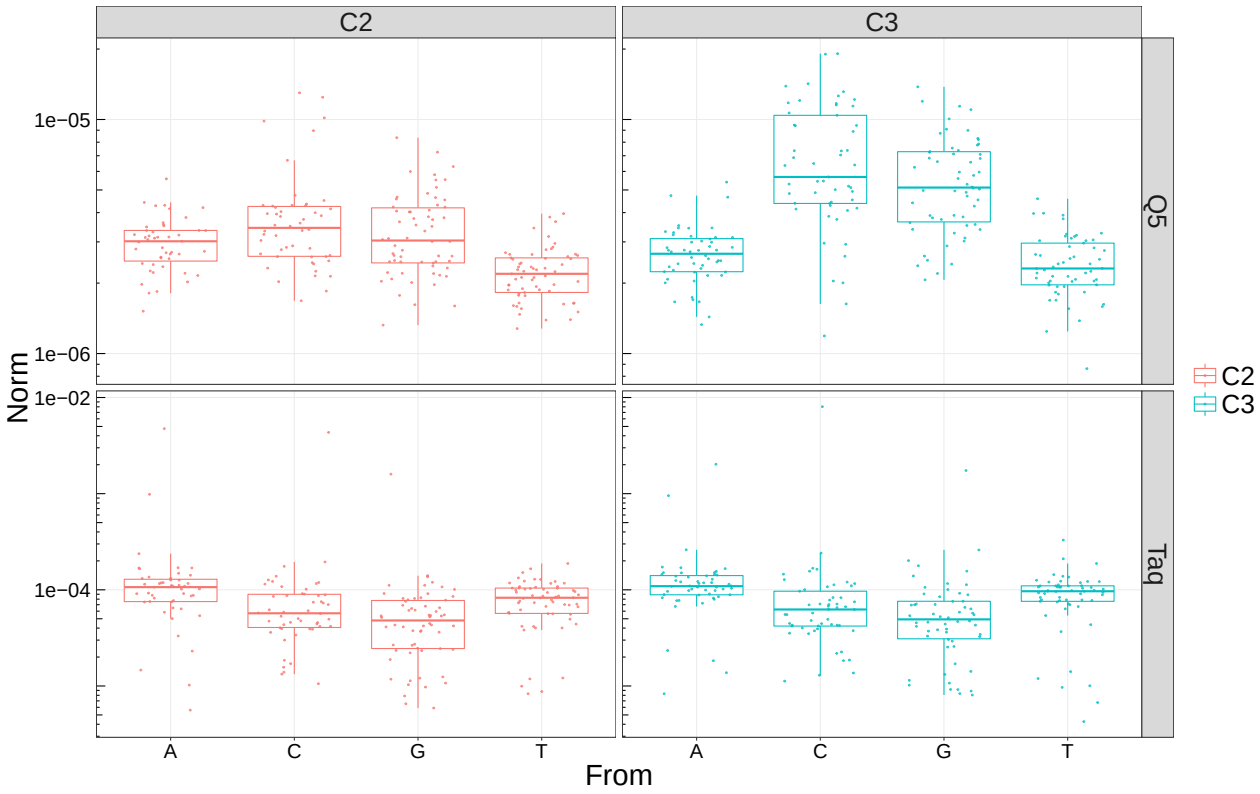


Mismatch Preference

```
review.mm <- review %>%
  filter(Type == 'M') %>%
  mutate(
    To = str_sub(Diff, 2, 2),
    From = str_sub(Diff, 1, 1)
  ) %>%
  count(Sample, Pos, From) %>%
  ungroup() %>%
  mutate(
    Construct = str_extract(Sample, "C([23])"),
    Polymerase = str_extract(Sample, "-(.*)-") %>% str_replace_all("-", ""),
    Repeat = str_extract(Sample, "\\d$")
  ) %>%
  inner_join(rename(review.counts, From=Char), by = c('Construct', 'From')) %>%
  inner_join(readCounts, by = 'Sample') %>%
  group_by(Construct, Polymerase, From, Pos) %>%
  summarise(Norm = mean(n / Count / Reads)) %>%
  ungroup()

# plot it
review.mm %>%
  ggplot(aes(x=From, y=Norm, color=Construct)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(position = position_jitter(w=0.3), size = 0.5, alpha = 0.8) +
  facet_grid(Polymerase ~ Construct, scales='free_y') +
```

```
  scale_y_log10() +
  annotation_logticks(sides = 'l')
```




```
# significance testing for A/T or G/C broadly
review.mm %>%
  mutate(Class = if_else(From %in% c('A', 'T'), 'A/T', 'C/G')) %>%
  group_by(Construct, Polymerase, Class) %>%
  summarise(Median = median(Norm))
```

```
## Source: local data table [8 x 4]
## Groups: Construct, Polymerase
##
## # grouped_dt [8 × 4]
##   Construct Polymerase Class       Median
##       <chr>      <chr> <chr>        <dbl>
## 1        C2         Q5   A/T 2.483312e-06
## 2        C2         Q5   C/G 3.283200e-06
## 3        C2        Taq   A/T 8.991179e-05
## 4        C2        Taq   C/G 5.271457e-05
## 5        C3         Q5   A/T 2.460987e-06
## 6        C3         Q5   C/G 5.442935e-06
## 7        C3        Taq   A/T 1.017910e-04
## 8        C3        Taq   C/G 5.506043e-05
```

```
review.mm %>%
  mutate(Class = if_else(From %in% c('A', 'T'), 'A/T', 'C/G')) %>%
  group_by(Construct, Polymerase) %>%
```

```
  do(tidy(wilcox.test(Norm ~ Class, data=.))) %>%
  select(-method)
```

```
## Source: local data table [4 x 5]
## Groups: Construct, Polymerase
##
## # grouped_dt [4 × 5]
##   Construct Polymerase statistic      p.value alternative
##       <chr>      <chr>     <dbl>        <dbl>      <fctr>
## 1        C2         Q5      3423 2.688707e-08   two.sided
## 2        C2        Taq      8596 6.759863e-08   two.sided
## 3        C3         Q5      1124 1.745020e-25   two.sided
## 4        C3        Taq      8940 9.220980e-10   two.sided
```

```
# significance testing for all pairwise comparisons
review.mm %>%
  group_by(Construct, Polymerase, From) %>%
  summarise(Median = median(Norm))
```

```
## Source: local data table [16 x 4]
## Groups: Construct, Polymerase
##
## # grouped_dt [16 × 4]
##    Construct Polymerase  From       Median
##        <chr>      <chr> <chr>        <dbl>
## 1         C2         Q5     A 3.018137e-06
## 2         C2         Q5     C 3.444028e-06
## 3         C2         Q5     G 3.040675e-06
## 4         C2         Q5     T 2.191338e-06
## 5         C2        Taq     A 1.066831e-04
## 6         C2        Taq     C 5.702962e-05
## 7         C2        Taq     G 4.813707e-05
## 8         C2        Taq     T 8.271209e-05
## 9         C3         Q5     A 2.669264e-06
## 10        C3         Q5     C 5.681917e-06
## 11        C3         Q5     G 5.119291e-06
## 12        C3         Q5     T 2.309372e-06
## 13        C3        Taq     A 1.091878e-04
## 14        C3        Taq     C 6.245210e-05
## 15        C3        Taq     G 4.933310e-05
## 16        C3        Taq     T 9.667671e-05
```

```
review.mm %>%
  group_by(Construct, Polymerase) %>%
  do(with(., tidy(pairwise.wilcox.test(Norm, From)))) %>%
  print(n=24)
```

```
## Source: local data table [24 x 5]
## Groups: Construct, Polymerase
##
## # grouped_dt [24 × 5]
##    Construct Polymerase group1 group2      p.value
```

```
##        <chr>      <chr> <fctr>  <chr>        <dbl>
## 1         C2         Q5      C      A 5.637903e-02
## 2         C2         Q5      G      A 4.811827e-01
## 3         C2         Q5      T      A 7.247495e-07
## 4         C2         Q5      G      C 4.811827e-01
## 5         C2         Q5      T      C 8.396410e-10
## 6         C2         Q5      T      G 2.977185e-07
## 7         C2        Taq      C      A 9.147524e-04
## 8         C2        Taq      G      A 1.292735e-06
## 9         C2        Taq      T      A 2.929496e-02
## 10        C2        Taq      G      C 6.622103e-02
## 11        C2        Taq      T      C 6.622103e-02
## 12        C2        Taq      T      G 1.389000e-04
## 13        C3         Q5      C      A 9.514120e-12
## 14        C3         Q5      G      A 4.455967e-12
## 15        C3         Q5      T      A 8.484781e-02
## 16        C3         Q5      G      C 8.742388e-02
## 17        C3         Q5      T      C 3.152033e-13
## 18        C3         Q5      T      G 3.862046e-14
## 19        C3        Taq      C      A 2.405056e-05
## 20        C3        Taq      G      A 2.369289e-07
## 21        C3        Taq      T      A 1.866667e-02
## 22        C3        Taq      G      C 1.423522e-01
## 23        C3        Taq      T      C 8.019580e-03
## 24        C3        Taq      T      G 9.509039e-05
```

Mismatch Hotspots

```
review %>%
  filter(Type == 'M') %>%
  count(Sample, Pos, Diff) %>%
  ungroup() %>%
  inner_join(readCounts, by = 'Sample') %>%
  mutate(
    Norm = n / Reads, Construct = str_extract(Sample, "C([023])"),
    Polymerase = str_extract(Sample, "-(.*)-") %>% str_replace_all("-", ""),
    Repeat = str_extract(Sample, "\\d$")
    ) %>%
  filter(Polymerase == 'Taq') %>%
  tbl_df %>%
  group_by(Sample) %>%
  top_n(n=6, wt = Norm) %>%
  ungroup() %>%
  filter(Construct == 'C2') %>%
  arrange(-Norm)
```

```
## # A tibble: 12 × 10
##      Sample   Pos  Diff     n  Reads   Errs       Norm Construct Polymerase Repeat
##       <chr> <int> <chr> <int>  <int>  <int>      <dbl>     <chr>      <chr>  <chr>
## 1  C2-Taq-1   101    AG 18863 166833 146829 0.11306516        C2        Taq      1
## 2  C2-Taq-2   101    AG 23844 215939 190403 0.11042007        C2        Taq      2
## 3  C2-Taq-1   181    CT 17503 166833 146829 0.10491330        C2        Taq      1
```

```
## 4  C2-Taq-2   181    CT 22181 215939 190403 0.10271882        C2        Taq      2
## 5  C2-Taq-1   101    AC 13621 166833 146829 0.08164452        C2        Taq      1
## 6  C2-Taq-2   101    AC 17249 215939 190403 0.07987904        C2        Taq      2
## 7  C2-Taq-2   160    GA 16045 215939 190403 0.07430339        C2        Taq      2
## 8  C2-Taq-1   160    GA 12214 166833 146829 0.07321094        C2        Taq      1
## 9  C2-Taq-1   181    CA 12117 166833 146829 0.07262952        C2        Taq      1
## 10 C2-Taq-2   181    CA 15552 215939 190403 0.07202034        C2        Taq      2
## 11 C2-Taq-2   181    CG  9174 215939 190403 0.04248422        C2        Taq      2
## 12 C2-Taq-1   181    CG  6497 166833 146829 0.03894313        C2        Taq      1
```

Polymerase Slippage

```
review %>%
  filter(Type == 'S') %>%
  count(Sample, Pos, Diff) %>%
  ungroup() %>%
  inner_join(readCounts, by = 'Sample') %>%
  mutate(
    Norm = n / Reads, Construct = str_extract(Sample, "C([023])"),
    Polymerase = str_extract(Sample, "-(.*)-") %>% str_replace_all("-", ""),
    Repeat = str_extract(Sample, "\\d$")
    ) %>%
  filter(Polymerase == 'Taq', Construct == 'C2') %>%
  tbl_df %>%
  group_by(Sample) %>%
  top_n(n=6, wt = Norm) %>%
  ungroup() %>%
  arrange(-Norm)
```

```
## # A tibble: 12 × 10
##      Sample   Pos  Diff     n  Reads   Errs         Norm Construct Polymerase Repeat
##       <chr> <int> <chr> <int>  <int>  <int>        <dbl>     <chr>      <chr>  <chr>
## 1  C2-Taq-1    93   GGA   100 166833 146829 0.0005994018        C2        Taq      1
## 2  C2-Taq-2    93    GA   119 215939 190403 0.0005510816        C2        Taq      2
## 3  C2-Taq-2    93   GGA   114 215939 190403 0.0005279269        C2        Taq      2
## 4  C2-Taq-1    93    GA    69 166833 146829 0.0004135872        C2        Taq      1
## 5  C2-Taq-2    92    GG    84 215939 190403 0.0003889987        C2        Taq      2
## 6  C2-Taq-1    92    GG    58 166833 146829 0.0003476530        C2        Taq      1
## 7  C2-Taq-1   102   GGC    39 166833 146829 0.0002337667        C2        Taq      1
## 8  C2-Taq-2    94    GA    50 215939 190403 0.0002315469        C2        Taq      2
## 9  C2-Taq-2   102   GGG    39 215939 190403 0.0001806066        C2        Taq      2
## 10 C2-Taq-2   102   GGC    34 215939 190403 0.0001574519        C2        Taq      2
## 11 C2-Taq-1    94    GA    25 166833 146829 0.0001498504        C2        Taq      1
## 12 C2-Taq-1   102   GGG    23 166833 146829 0.0001378624        C2        Taq      1
```

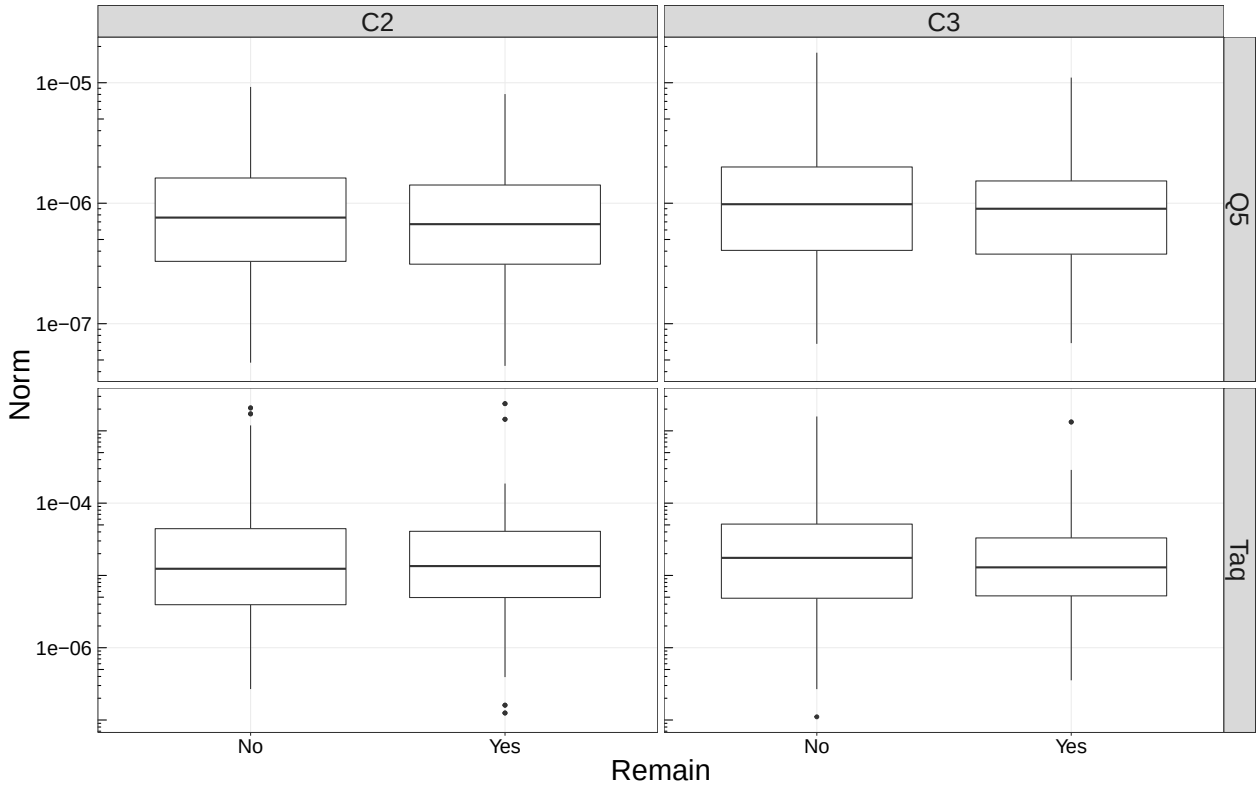
Are Mismatches the Last Base Snythesized

```
last.base <- c(c2, c3) %>%
  str_split("") %>%
  unlist() %>%
```

```
  data.table(
    LastBase = .,
    Construct = c(rep("C2", 220), rep("C3", 220)),
    Pos = rep(seq(2, 221), 2)
  )

last.base.test <- review.mm.subtype %>%
  select(-Class) %>%
  inner_join(last.base, by = c('Construct', 'Pos')) %>%
  mutate(
    From = str_sub(Diff, 1, 1),
    To = str_sub(Diff, 2, 2)
  ) %>%
  filter(!LastBase == From) %>%
  mutate(Remain = if_else(LastBase == To, 'Yes', 'No'))

# plot it
last.base.test %>%
  ggplot(aes(x=Remain, y=Norm)) +
  geom_boxplot() +
  facet_grid(Polymerase ~ Construct, scales='free_y') +
  scale_y_log10() +
  annotation_logticks(sides='l')
```




```
# doesnt look significant, but to be sure...
last.base.test %>%
  group_by(Construct, Polymerase) %>%
  do(tidy(wilcox.test(Norm ~ Remain, data=.)))
```

```
## Source: local data table [4 x 6]
## Groups: Construct, Polymerase
##
## # grouped_dt [4 × 6]
##   Construct Polymerase statistic    p.value                                                method alterna
##       <chr>      <chr>     <dbl>      <dbl>                                                 <fctr>      <
## 1        C2         Q5  118988.0 0.05968531 Wilcoxon rank sum test with continuity correction   two.s
## 2        C2        Taq  113850.0 0.69324449 Wilcoxon rank sum test with continuity correction   two.s
## 3        C3         Q5  130833.5 0.08541851 Wilcoxon rank sum test with continuity correction   two.s
## 4        C3        Taq  133898.0 0.12201621 Wilcoxon rank sum test with continuity correction   two.s
```