Seq QA

Nate Olson

September 21, 2016

Sequence Processing

Sequencing Data Quality Assessment

Source Data

Names of source data files and md5 sums for provenance checking under session info tab.

```
project_dir <- "~/Projects/16S_etec_mix_study/"
sav_dir <- paste0(project_dir,"data/basespace_sav/")
sav_list <- list(
    march2016_v3 = paste0(sav_dir,"run_18071097_sav"),
    basespace_v3 = paste0(sav_dir,"run_3861867_sav"),
    april2016_v3 = paste0(sav_dir,"run_18527531_sav")
)

# sav_list %>%
#     map(list.files,full.names = TRUE, recursive = TRUE) %>%
#     flatten_chr() %>% md5table()
```

```
fastq_dir <- paste0(project_dir, "data/fastq/jhu_run2/")
dat_files <- list.files(path = fastq_dir,pattern = "001.fastq.gz$",
                            recursive = TRUE, full.names = TRUE)
seq_ds_id <- dat_files %>% str_split("/") %>% flatten_chr() %>%
    grep(pattern = "fastq.gz", .,value = TRUE) %>%
    str_replace(".fastq.gz","") %>% paste0("Fq_",.) %>%
    str_replace("-",".")
names(dat_files) <- seq_ds_id
# md5_tbl <- dat_files %>% md5table()
```

Error Rate

```
error_df <- sav_list %>% map(savR) %>%
    map_df(errorMetrics,.id = "ds") %>%
    separate(ds, c("ds","chemistry"),sep = "_") %>%
    mutate(read_len = if_else(chemistry == "v3",301, 251),
           read = if_else(cycle < read_len, "R1","R2"))
```

QA Metrics

To generate summaries of QA metrics for the 384 datasets in the study (192 samples with forward and reverse reads) used Rqc to calculate the quality metrics, extracted the resulting data and generated summary plots more suitable to presenting QA data for large numbers of datasets.

Short Read package also has functions for extracting QA results and generating plots that could be used to supplement or inplace of the following Rqc based analysis.

- Accessor functions for ShortRead package QA (not exported)

- Run Summary : .ppnCount, .df2a, .laneLbl, .plotReadQuality
- Read Distribution : .plotReadOccurrences, .freqSequences
- Cycle Specific : .plotCycleBaseCall, .plotCycleQuality
- Tile Performance : .atQuantile, .colorkeyNames, .plotTileLocalCoords, .tileGeometry, .plotTileCounts, .plotTileQualityScore
- Alignment : .plotAlignQuality
- Multiple Alignment : .plotMultipleAlignmentCount
- Depth of Coverage : .plotDepthOfCoverage
- Adapter Contamination : .ppnCount

QA Data

Issues with running rqcQA on files within knitr and R/ Rstudio in general. Issue with unused connections not closing. Rqc `rqcQC-method` closes the connection but it remains open which throws an error when processing large numbers of files (no error for 96 but error for 160). Spliting the input list into sets of 6 with 6 parallel workers (`rqcQA` uses BiocParallel's `bpmapply` function for parallelization) avoids does not throw an error. Error potentially due to gzipped files opening two connections and RqcA only closes one of the connection. Generated R script `run_rqca.R` to generate and save a list with the rqcA output.

Code from analysis that was moved to the script.

```
qa_list <- list()
step_size <- 6 #for all data

n_files <- length(dat_files)
for(i in 0:((n_files/step_size) - 1)){
    print(i)
    qa_list <- c(qa_list,
                      rqcQA(dat_files[(step_size * i+1):(step_size*(i + 1))],
                          group = read_groups[(step_size *i+1):(step_size *(i+1))],
                          workers = step_size))
}

names(qa_list) <- names(dat_files)
names(read_groups) <- names(qa_list)
```

Loading qa_list from rds created by script

```
# from script but not saved
read_groups <- rep(NA,length(dat_files))
read_groups[grepl("/1-.*_R1", dat_files)] <- "plate1_R1"
read_groups[grepl("/1-.*_R2", dat_files)] <- "plate1_R2"
read_groups[grepl("/2-.*_R1", dat_files)] <- "plate2_R1"
read_groups[grepl("/2-.*_R2", dat_files)] <- "plate2_R2"

qa_list <- readRDS("../data/rqcQA_list.rds")
```

Tidying QA data

Metadata

```
## Run
grp_df <- data_frame(read_group = read_groups,
                     seq_ds_id,
                     filename = basename(dat_files)) %>%
    separate(read_group,c("plate","Read")) %>%
    mutate(ill_id = str_replace(filename, "_.*",""))

## read count data
qa_file_info <- perFileInformation(qa_list) %>%
    select(-format,-path)

## study metadata
data(sample_sheet)
meta_df <- sample_sheet %>%
    mutate(pos_ns = str_replace(pos, "_",""),
           ill_id = paste(pcr_16S_plate, pos_ns, sep = "-")) %>%
    filter(seq_lab == "JHU", barcode_lab == "JHU") %>%
    mutate(pcr_16S_plate = as.character(pcr_16S_plate)) %>%
    left_join(grp_df) %>% left_join(qa_file_info)
```

```
## Joining, by = "ill_id"
```

```
## Joining, by = "filename"
```

```
## Warning in left_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## factor and character vector, coercing into character vector
```

Read Level Metrics

```
qa_read_df <- qa_list %>% map_df(perReadWidth) %>% left_join(meta_df) %>%
    mutate(len_prop = count/reads)
```

```
## Warning in bind_rows_(x, .id): Unequal factor levels: coercing to character
```

```
## Joining, by = "filename"
```

```
qa_read_q_df <- qa_list %>% map_df(perReadQuality) %>% left_join(meta_df)
```

```
## Warning in bind_rows_(x, .id): Unequal factor levels: coercing to character
```

```
## Warning in bind_rows_(x, .id): Unequal factor levels: coercing to character
```

```
## Joining, by = c("filename", "group")
```

```
qa_read_freq <- qa_list %>% map_df(perReadFrequency) %>% left_join(meta_df)
```

```
## Warning in bind_rows_(x, .id): Unequal factor levels: coercing to character
```

```
## Joining, by = "filename"
```

Cycle Level Metrics

```
## amplicon position
amp_pos_df <- data_frame(cycle = rep(1:300, 2),
                         Read = rep(c("R1","R2"), each = 300),
                         amp_pos = c(1:300,c(460 - 1:300)))

qa_cycle_q_df <- qa_list %>% map_df(perCycleQuality) %>%
    as_data_frame() %>%
```

```
    mutate(cycle = as.numeric(as.character(cycle))) %>%
    filter(count != 0) %>% # not sure if this impacts the smoothing function ...
    left_join(meta_df) %>% left_join(amp_pos_df)
```

```
## Warning in bind_rows_(x, .id): Unequal factor levels: coercing to character
```

```
## Warning in bind_rows_(x, .id): Unequal factor levels: coercing to character
```

```
## Joining, by = c("filename", "group")
```

```
## Joining, by = c("cycle", "Read")
```

```
# qa_cycle_base_df <- qa_list %>% map_df(perCycleBasecall) %>% as_data_frame() %>%
    # mutate(cycle = as.numeric(as.character(cycle))) %>%
    # left_join(meta_df) %>% left_join(amp_pos_df)
## Additional Cycle level metrics
# qa_cycle_q_avg <- rqcCycleAverageQualityCalc(qa_list)
# qa_cycle_bc_avg <- rqcCycleBaseCallsCalc(qa_list)
# qa_cycle_GC <- rqcCycleGCCalc(qa_list) %>% as_data_frame()
```

Read Counts

Two barcoded experimental sample has less than 50,000 reads. The rest of the samples with less than 50,000 reads are negative PCR controls (NTC).

```
meta_df %>% mutate(exp_ntc = ifelse(sampleID == "NTC", "NTC","EXP")) %>%
    group_by(exp_ntc, Read, plate) %>%
    summarise(mean_lib_size = mean(reads),
              min_lib_size = min(reads),
              median = median(reads),
              max_lib_size = max(reads)) %>%
    kable(caption = "Summary statistics for experimental and no template control samples by PCR plate a
```

Table 1: Summary statistics for experimental and no template control samples by PCR plate and read.

exp_ntc	Read	plate	mean_lib_size	min_lib_size	median	max_lib_size
EXP	R1	plate1	88412.82	3195	85977.0	152267
EXP	R1	plate2	96263.40	46328	97615.5	143110
EXP	R2	plate1	88412.82	3195	85977.0	152267
EXP	R2	plate2	96263.40	46328	97615.5	143110
NTC	R1	plate1	3698.50	1305	1715.0	13146
NTC	R1	plate2	13159.83	5216	9750.5	25349
NTC	R2	plate1	3698.50	1305	1715.0	13146
NTC	R2	plate2	13159.83	5216	9750.5	25349

```
meta_df %>%
    ggplot() +
        geom_histogram(aes(x = reads), position = "dodge") +
        facet_grid(plate~Read) + scale_x_log10() +
        geom_rug(aes(x = reads, color = sampleID)) +
        geom_vline(aes(xintercept = 50000), color = "grey80") +
        theme_bw() + labs(x = "Library Size (log10)",
                          y = "Count", color = "Sample ID") +
```

```
        theme(legend.position = "bottom")
```

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

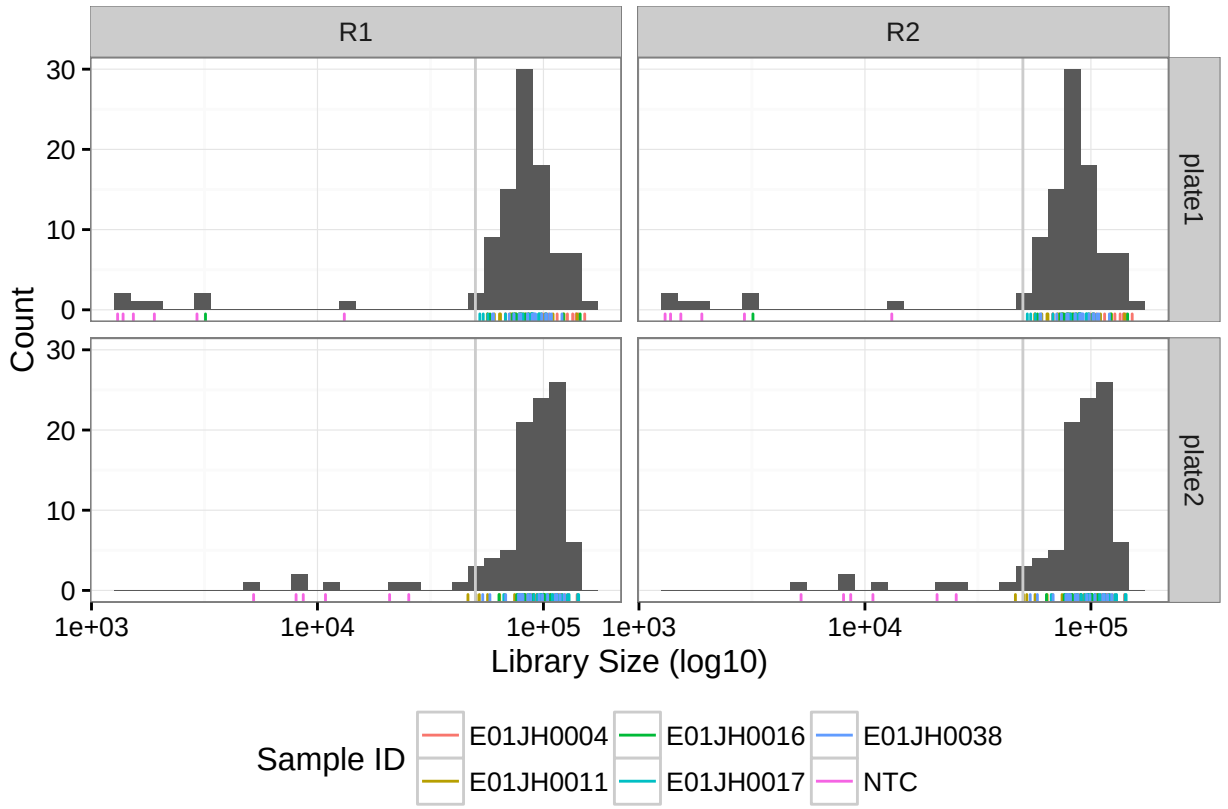

Figure 1: Number of reads per barcoded sample (Library Size), by read direction (X-facet) and replicate 16S PCR plate (Y-facet). Vertical line indicates 50,000 reads per barcoded sample.

```
meta_df  %>% filter(sampleID != "NTC",reads < 50000) %>%
    select(sampleID, dilution, pos, plate, Read, reads) %>%
    spread(Read, reads) %>%
    kable(caption = "Barcoded experimental samples with less than 50,000 reads")
```

Table 2: Barcoded experimental samples with less than 50,000 reads

sampleID	dilution	pos	plate	R1	R2
E01JH0011	3	D_2	plate2	46328	46328
E01JH0016	5	F_9	plate1	3195	3195

Read Length Distribution

Excluding negative control samples (NTC) only one sample had > 2.5% of the reads in the dataset less than 290 bp. Biosample E01JH0016 dilution 5 from plate 1 replicates had greater than 10% of reads < 290 bp in length.

code for calculating proportions

```
qa_read_len_bin <- qa_read_df %>%
    mutate(size_bin = ifelse(width > 290, ">290","<290")) %>%
    group_by(filename, id, sampleID, plate, dilution, size_bin,Read) %>%
    summarise(size_prop = sum(len_prop))
```

```
qa_read_len_bin %>% filter(size_bin == "<290") %>%
    ggplot() +
        geom_histogram(aes(x = size_prop)) +
        geom_rug(aes(x = size_prop, color = sampleID)) +
        theme_bw() +
        labs(x = "Proportion of reads < 290 bp", color = "Sample ID")
```

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

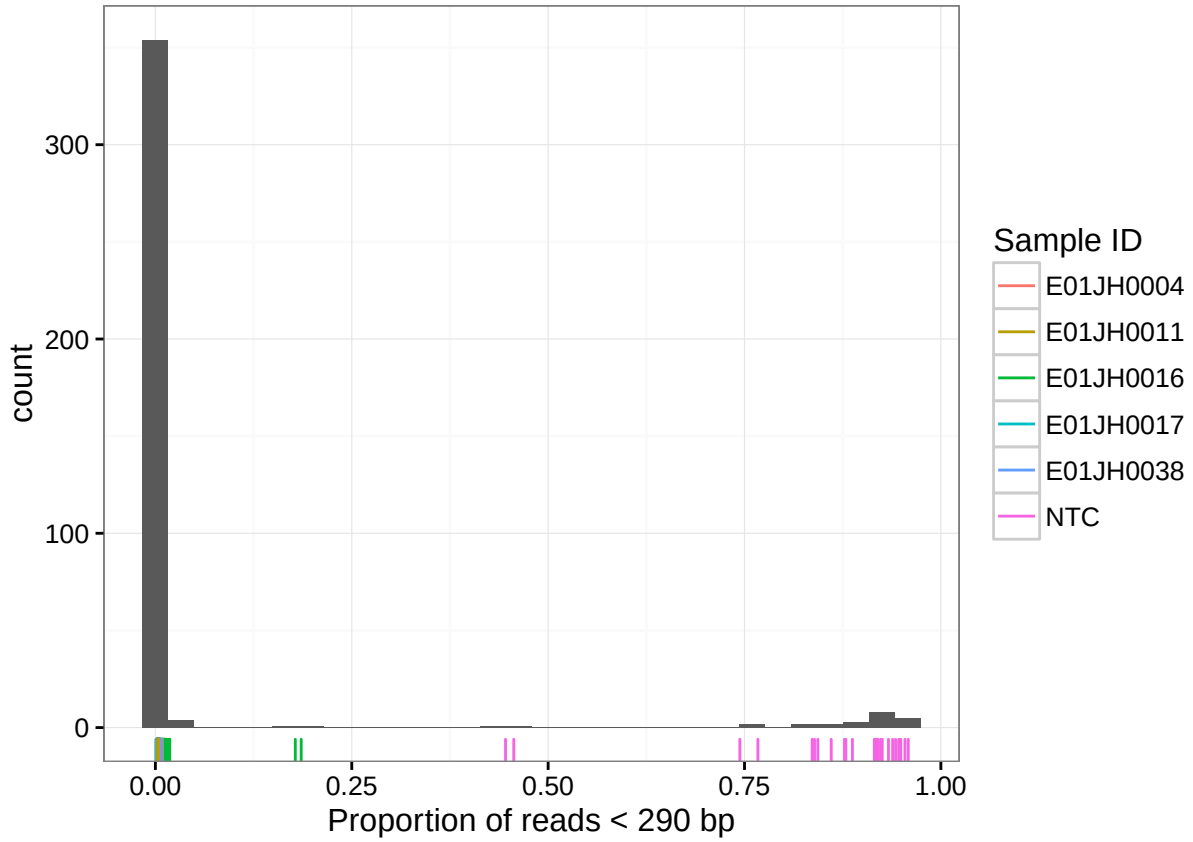

Figure 2: Histogram of the proportion of reads less than 290 bp for per barcoded sample.

For the experimental sample with $> 10\%$ of read shorter than 290 bp the shorter reads are 166 bp which is too short to generate merge contigs from forward and reverse reads.

```
short_samples <- qa_read_len_bin %>%
    filter(size_bin == "<290" & size_prop > 0.10,
           sampleID != "NTC") %>% .$filename
```

```
qa_read_df %>% filter(filename %in% short_samples) %>%
    ggplot() + geom_bar(aes(x = as.factor(width),
                            y = count, fill = Read),
                        stat="identity",position = "dodge") +
```

```
    theme_bw() +
    labs(x = "Read Length", y = "Number of Reads", fill = "Direction")
```

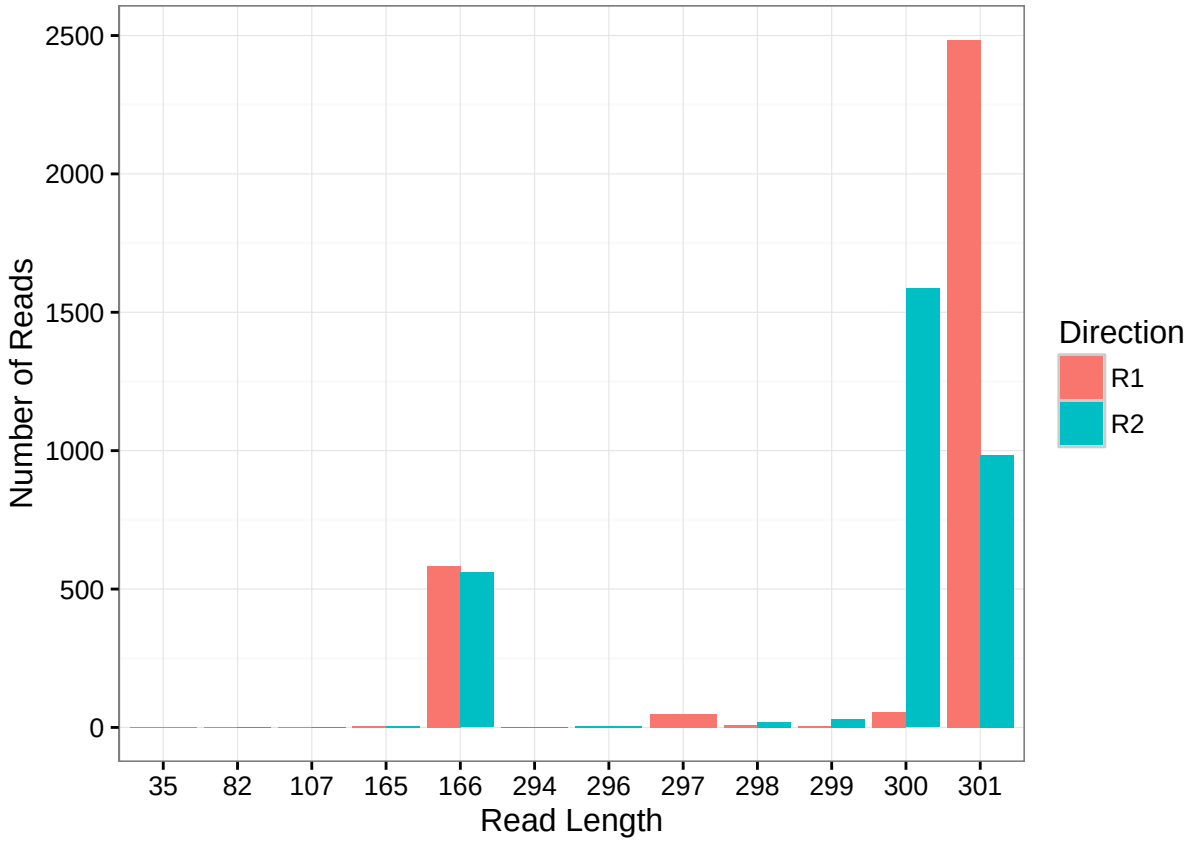

Figure 3: Distribution of read lengths for E01JH0016 dilution 5 plate 1. X-axis is ploted on a discrete not continuous scale.

PhiX Error Rate

The sequencing error rate data was obtained from the Basespace sequencing run report downloaded from Basespace (SAV file). Error rate is compared to the first sequencing run and a 16S public dataset on basespace (16S-Metagenomic-Library-Prep run id 3861867). The error rate for the second run was lower for both R1 and R2 compared to the first run but still higher than the error rate for the public dataset.

```
ggplot(error_df) +
    geom_point(aes(x = cycle, y = errorrate, color = ds),
               alpha = 0.01) +
    geom_smooth(aes(x = cycle, y = errorrate, color = ds)) +
    theme_bw() +
    labs(x = "Cycle", y = "PhiX Error Rate", color = "Dataset") +
    facet_grid(chemistry~read, scales = "free_x")
```

Base Quality Score

Read BQ

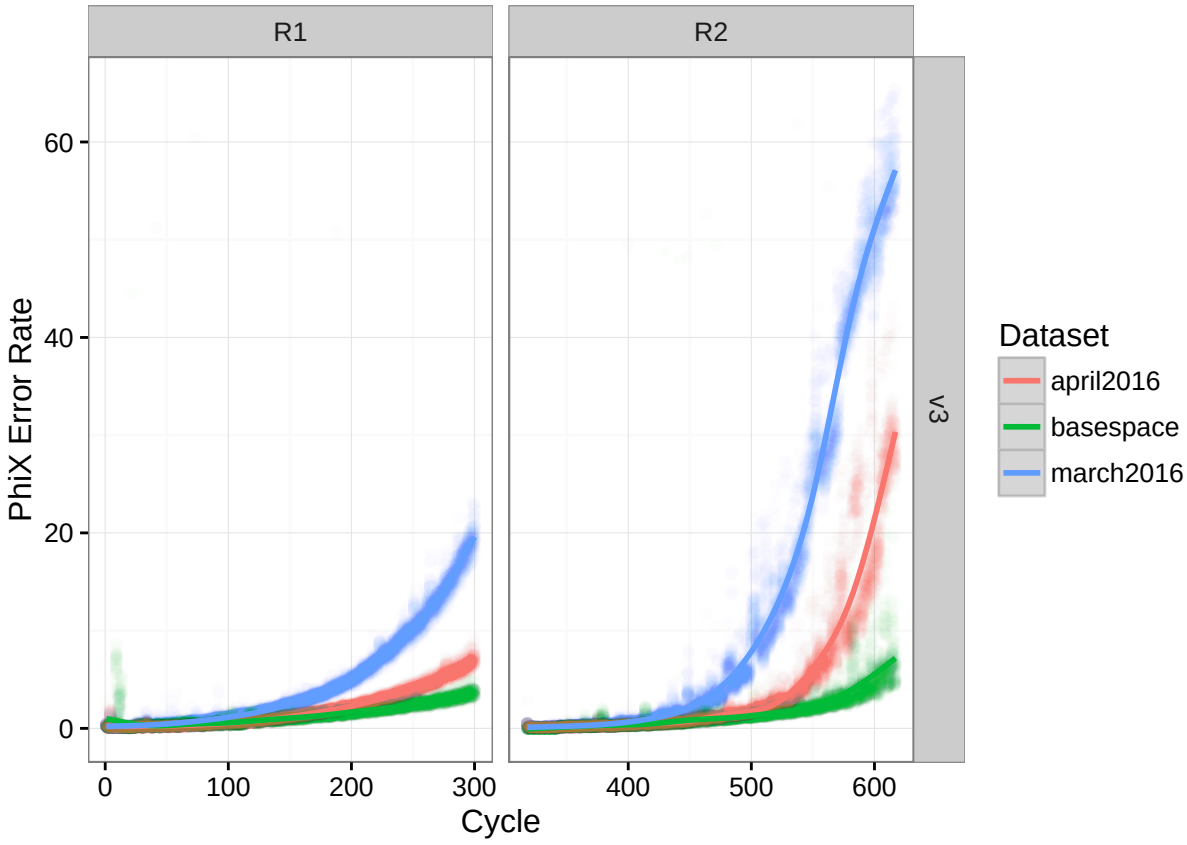

Figure 4: PhiX error rate for initial and reseuqencing of JHU barcoded samples compared to the public dataset.

Differences in forward and reverse read average base quality score distributions consistent between replicate plates. A distinct population of barcoded datasets, NTC vs experimental samples, with a higher proportion of lower base quality scores for forward read datasets. For reverse reads the popultion of datasets with lower base quality scores is more hetergeneous.

```
qa_read_q_df %>%
    ggplot() + geom_density(aes(x = average, fill = sampleID,
                                group = filename), alpha = 0.25) +
    facet_grid(plate~Read) + theme_bw() +
    labs(x = "Read Average Base Quality Score", y = "Density", fill = "Sample ID") +
    theme(legend.position = "bottom")
```

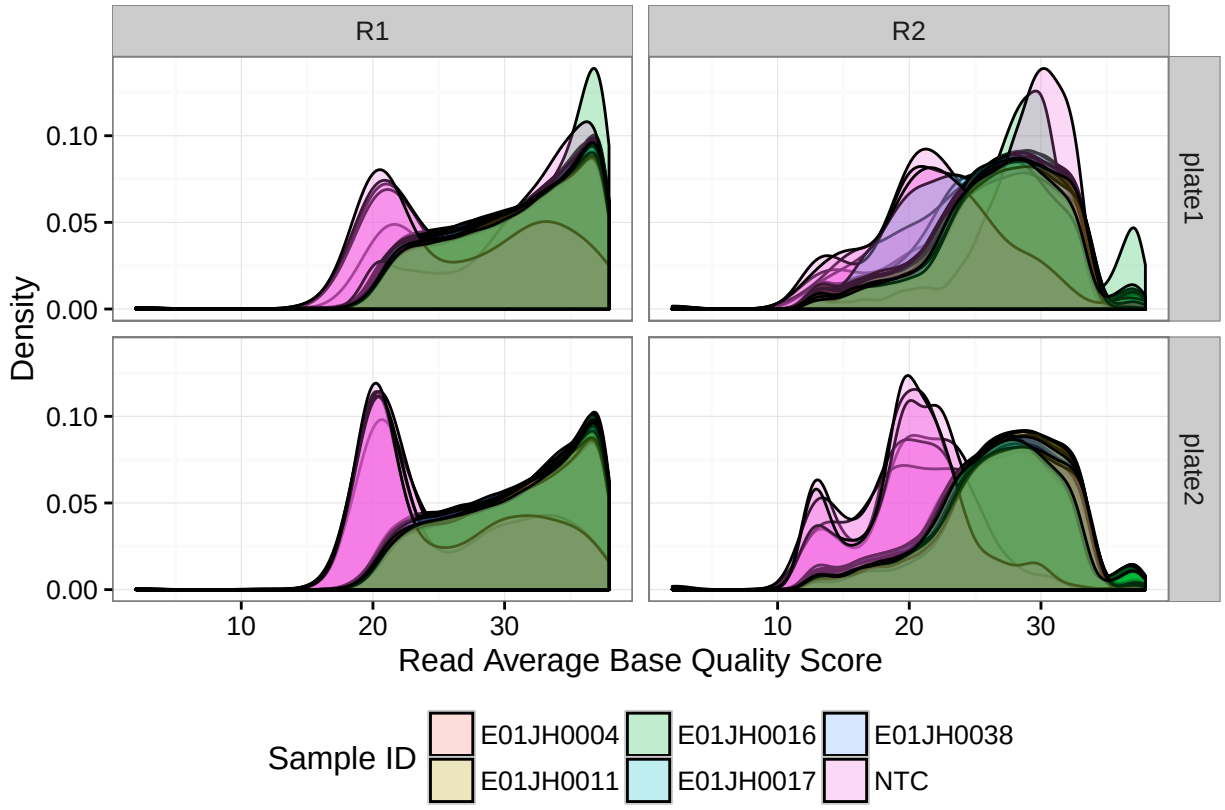

Figure 5: Distribution of base quality scores per barcoded samples.

Cycle BQ

Similar to the overall quality score distributions, the NTC quality score by sequencing cycle is consistently lower quality compared to the experimental samples. Cycle base quality score is more homogeneous from PCR plate 2 samples than plate 1.

```
qa_cycle_q_df %>%
    ggplot(aes(x = cycle, y = score)) +
    geom_smooth(aes(weight = count, color = sampleID, group = seq_ds_id)) +
    facet_grid(plate~Read) + theme_bw() +
    labs(x = "Sequencing Cycle", y = "Base Quality Score", color = "Sample ID")
```

Base quality score distribution across datasets for R1 and R2 reads by amplicon position. This is not a read level analysis but average quality score for individual barcoded datasets.

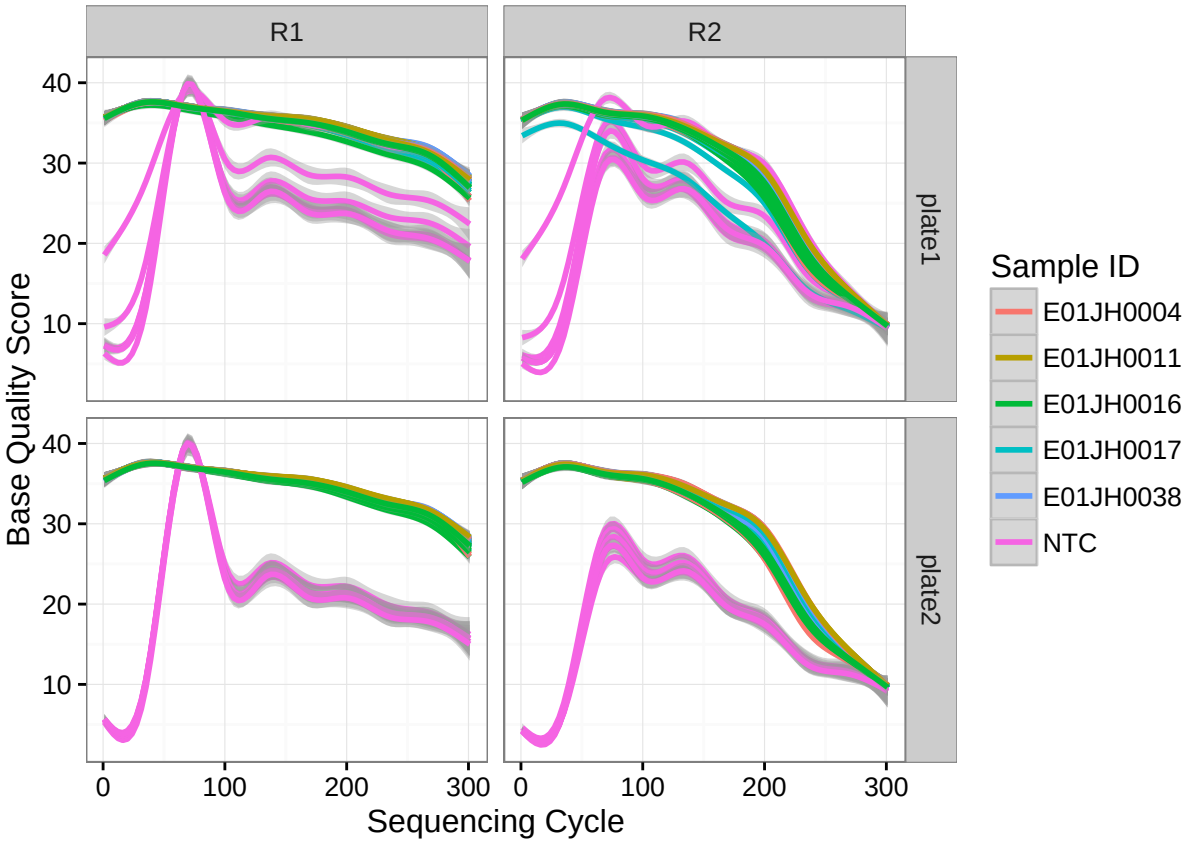

Figure 6: Smoothing spline of the base quality score by sequencing cycle.

```
qa_cycle_q_df %>% ggplot(aes(x = amp_pos, y = score)) +
    geom_vline(aes(xintercept = 300), color = "grey60") +
    geom_vline(aes(xintercept = 160), color = "grey60") +
    geom_smooth(aes(weight = count, color = sampleID, group = filename)) +
    facet_grid(plate~.) + theme_bw() +
    scale_x_continuous(breaks = c(0,150,300, 450)) +
    labs(x = "Amplicon Position",
         y = "Base Quality Score", color = "Sample ID")
```

```
## Warning: Removed 12036 rows containing non-finite values (stat_smooth).
```

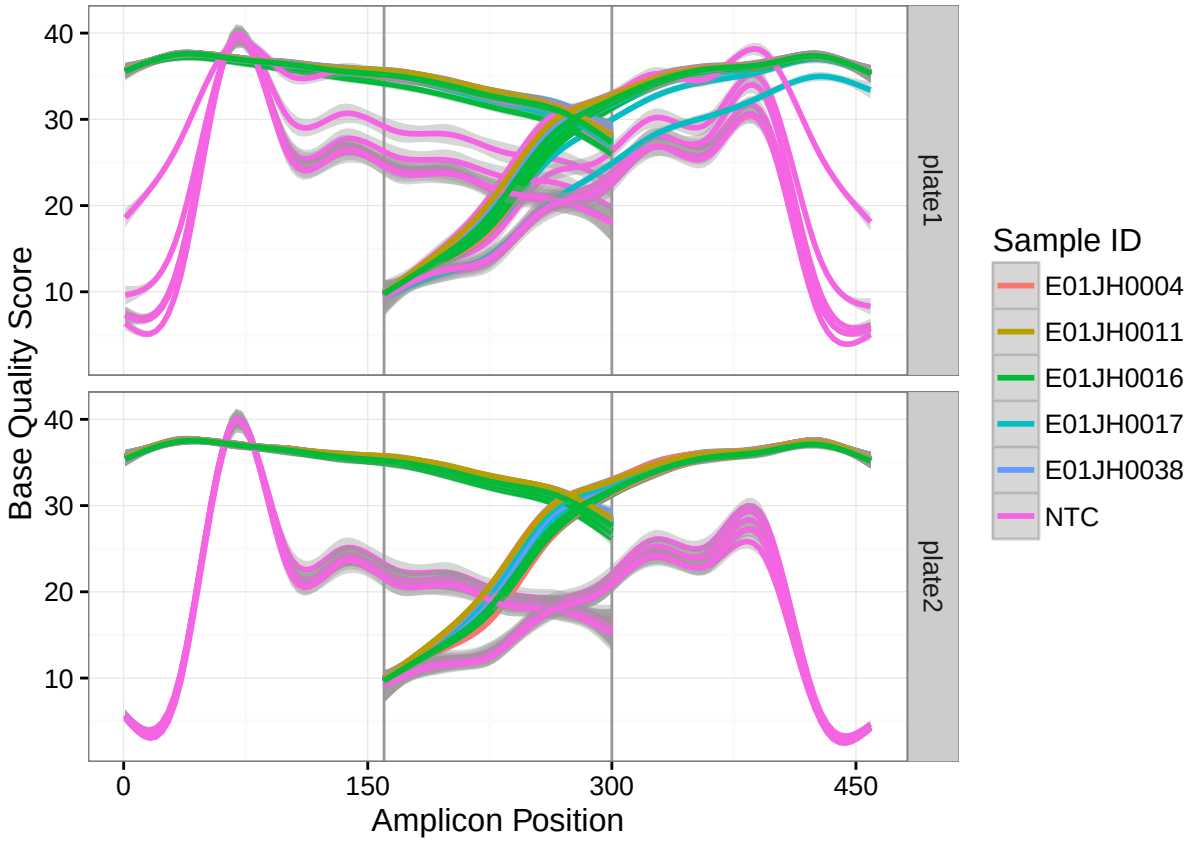

Figure 7: Smoothing spline of the base quality score by sequencing cycle. Vertical lines indicate approximate overlap region between forward and reverse reads.

The quality scores for the two plates are consistent for most samples . A few other samples had lower base quality scores across the amplicon. PCR plate 1 E01JH0017 dilution 4 samples have lower average quality, for amplicon positions > 200 bp.

```
qa_cycle_q_df %>% filter(sampleID !="NTC") %>%
    ggplot(aes(x = amp_pos, y = score)) +
    geom_smooth(aes(weight = count, color = plate, group = filename)) +
    facet_grid(dilution~sampleID) + theme_bw() +
    labs(x = "Amplicon Position", y = "Base Quality Score", color = "Sample ID") +
    theme(legend.position = "bottom")
```

```
## Warning: Removed 11462 rows containing non-finite values (stat_smooth).
```

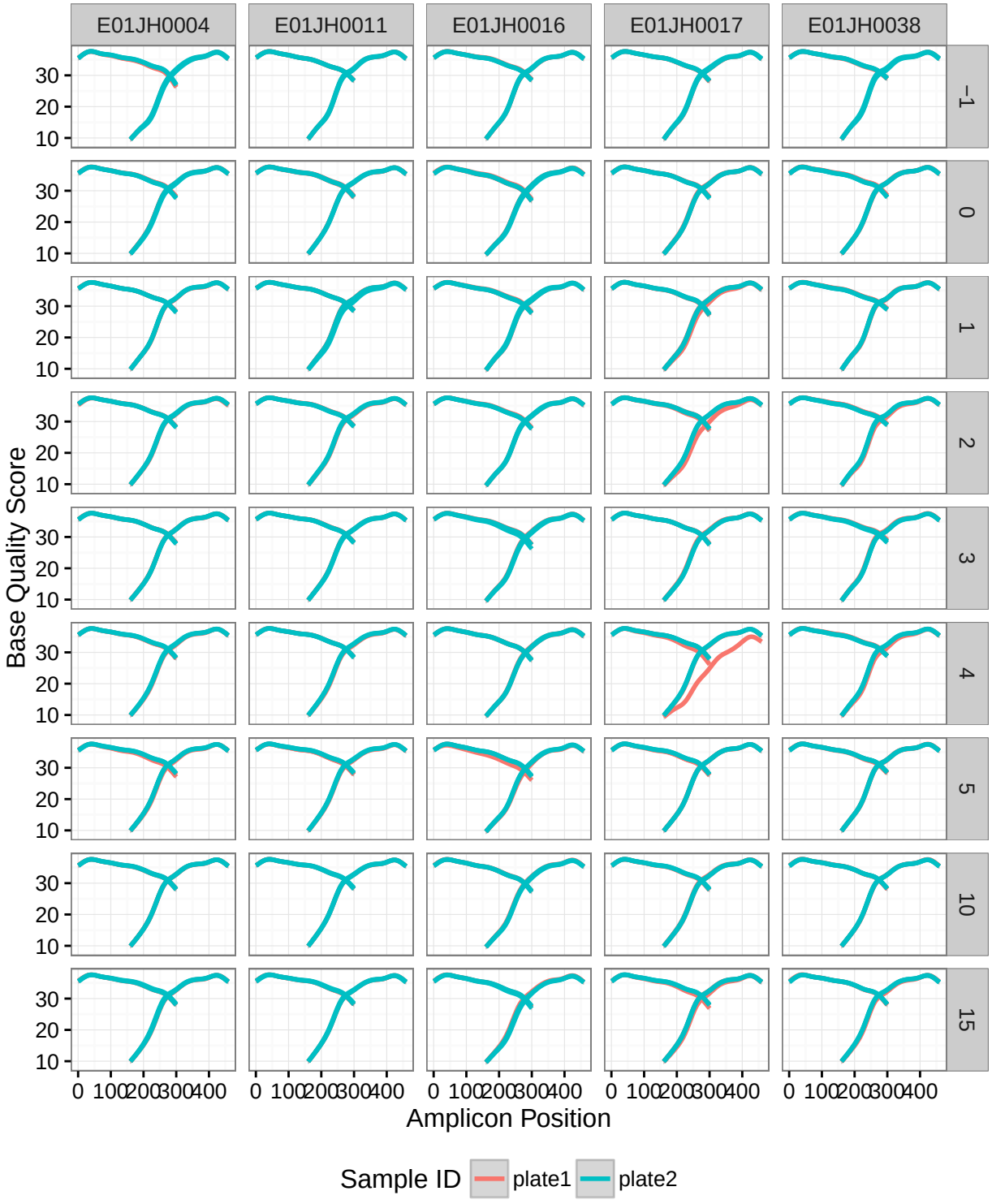

Figure 8: Smoothed average of the base quality score across the amplicon faceted by biological sample ID (X-axis) and dilution (Y-axis)