# supernova 16s Notebook

```
library(dada2)
```

```
## Loading required package: Rcpp
```

```
library(decontam)
library(ggplot2)
library(jsonlite)
library(lattice)
library(reshape)
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
PROJECT_DIR = "/home/rpetit/projects/supernova-16s"
setwd(PROJECT_DIR)
```

## Input FASTQs

Input FASTQs had adapters removed using BBDuk prior to DADA2 analysis. This was accomplished with the following command:

The adapter cleaned FASTQ files were stored in `data/adapter-cleaned`

## Dada2 Analysis

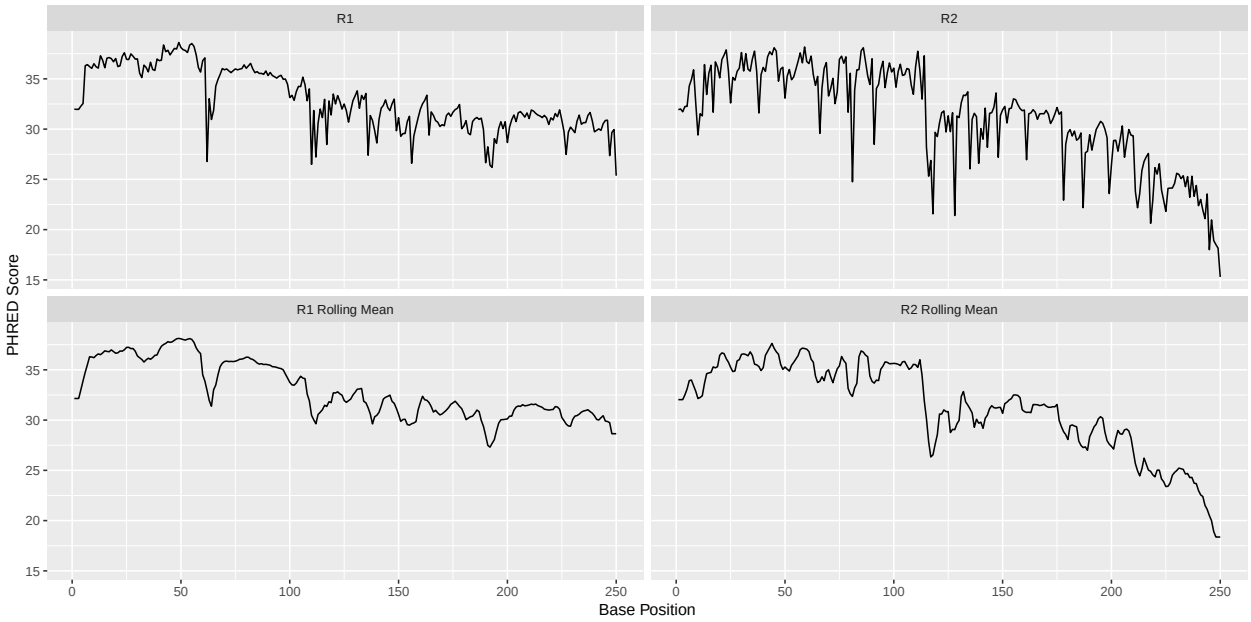
### Setting Up Files and Naming

```
# Input FASTQs
fastq_path = paste0(PROJECT_DIR, "/data/adapter-cleaned")
fastq_r1 <- sort(
  list.files(fastq_path, pattern="_R1.fastq.gz", full.names = TRUE)
)
fastq_r2 <- sort(
  list.files(fastq_path, pattern="_R2.fastq.gz", full.names = TRUE)
)
sample_names <- sapply(strsplit(basename(fastq_r1), "_"), `[`, 1)
```

### Plot Project FASTQ Quality Profile

```
json_r1 <- sort(list.files(fastq_path, pattern="_R1.json", full.names = TRUE))
json_r2 <- sort(list.files(fastq_path, pattern="_R2.json", full.names = TRUE))
r1_profile = NULL
```

```
r2_profile = NULL
for (i in 1:length(json_r1)){
  r1_df = as.data.frame(fromJSON(txt=json_r1[i])$per_base_quality)
  r2_df = as.data.frame(fromJSON(txt=json_r2[i])$per_base_quality)
  if (is.null(r1_profile)) {
    r1_profile = r1_df
    r2_profile = r2_df
  }
  else {
    r1_profile = rbind(r1_profile, r1_df)
    r2_profile = rbind(r2_profile, r2_df)
  }
}

colnames(r1_profile) <- as.integer(gsub("X", "", colnames(r1_profile)))
colnames(r2_profile) <- as.integer(gsub("X", "", colnames(r2_profile)))
means = data.frame(
  position=1:ncol(r1_profile),
  R1=as.data.frame(colMeans(r1_profile))[,1],
  R2=as.data.frame(colMeans(r2_profile))[,1]
)
means$"R1 Rolling Mean" <- rollmean(means$R1, 5, fill="extend")
means$"R2 Rolling Mean" <- rollmean(means$R2, 5, fill="extend")
melted_vals = melt(
  means,
  id.vars = "position",
  measure.vars = c("R1", "R2", "R1 Rolling Mean", "R2 Rolling Mean")
)
p <- ggplot(melted_vals, aes(x=position, y=value)) +
  geom_line() +
  facet_wrap(~ variable, ncol=2) +
  xlab("Base Position") +
  ylab("PHRED Score")
print(p)
```

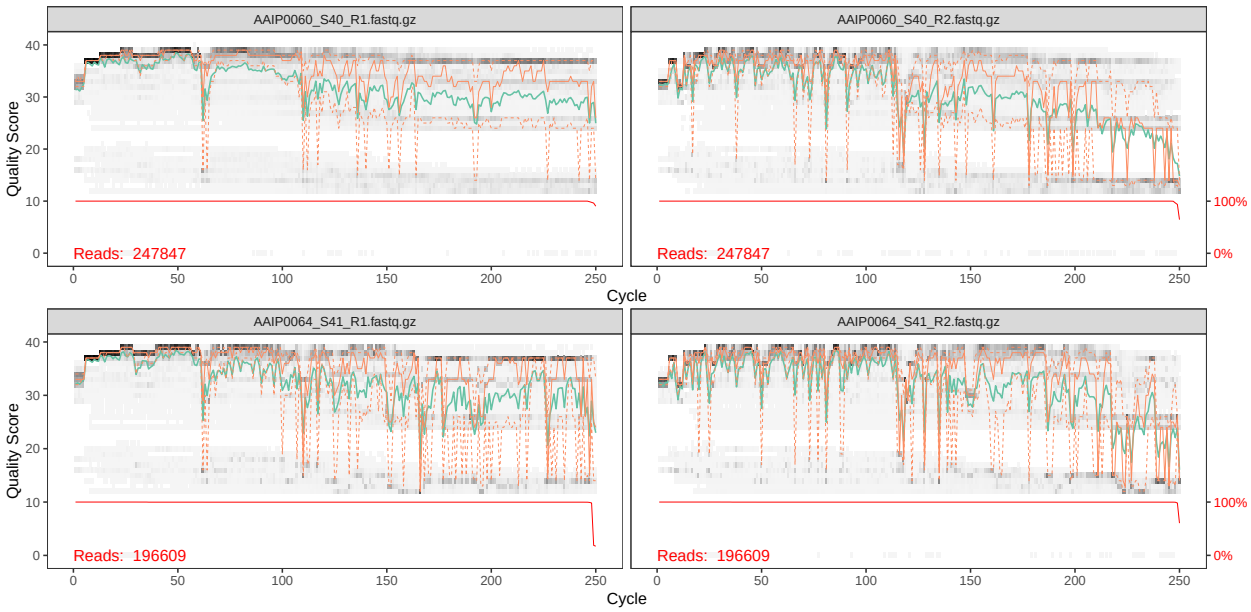



With these plots, we can get a general idea of the per-base mean quality across all samples in the project. There are two plots for forward (R1) and reverse (R2) reads. The top plots ("R1" and "R2") are the per-base mean quality across all samples, and the bottom plots ("R1 Rolling Mean" and "R2 Rolling Mean") use a rolling mean that is based on a centered 5bp sliding window.

As expected the quality tails off in the R2 reads, but overall the quality is good.

## Plot Sample FASTQ Quality Profiles (Subset)

```
for (i in 1:2) {
  print(plotQualityProfile(c(fastq_r1[i], fastq_r2[i])))
}
```



## Filter and Trim

For this project the V1 and V2 region of the 16s rRNA gene was sequenced. This region is roughly ~360bp and our project includes 2x250 Illumina MiSeq sequencing. For filtering and trimming we need to maintain a total length greater than 360bp plus a 20bp overlap, so ~380bp. Given 2x250bp reads, this gives 120bp to play with for trimming.

```
# Filter and Trim
filt_r1 <- file.path(
  PROJECT_DIR, "data", "filtered", paste0(sample_names, "_R1_filt.fastq.gz")
)
filt_r2 <- file.path(
  PROJECT_DIR, "data", "filtered", paste0(sample_names, "_R2_filt.fastq.gz")
)
names(filt_r1) <- sample_names
names(filt_r2) <- sample_names
out <- as.data.frame(
  filterAndTrim(fastq_r1, filt_r1, fastq_r2, filt_r2, truncLen=c(250,210),
                minLen=190, maxN=0, maxEE=c(2,5), truncQ=2, rm.phix=TRUE,
                compress=TRUE, multithread=TRUE)
)
filtered_stats <- out
colnames(filtered_stats) <- c("input", "filtered")
rownames(filtered_stats) <- sample_names
filtered_stats$percent_filtered_out <- 1 - (filtered_stats$filtered / filtered_stats$input)
```

Using the quality profile of the complete set of samples, the following parameters were using for fitering and trimming:

| Parameters | Description |
|---|---|
| `truncLen=c(250,210)` | Trim length of forward (250bp) and reverse (210bp) reads |
| `minLen=185` | Remove reads that are less than 190bp after trimming |
| `maxN=0` | Filter reads containing ambiguous (e.g. N) base calls |
| `maxEE=c(2,5)` | Filter reads with more than expected basecall errors (R1: 2, R2: 5) |
| `truncQ=2` | Trim reads at the position with a PHRED score of Q2 |
| `rm.phix=TRUE` | Remove any reads matching PhiX (Illumina control) |
| `compress=TRUE` | Gzip output FASTQs |
| `multithread=TRUE` | Use multiple processors |

Below is the top 10 samples with the most reads filtered out.

```
head(filtered_stats[order(filtered_stats$percent_filtered_out, decreasing=TRUE),], n=10)
```

```
##           input filtered percent_filtered_out
## AAIP0064 196609    17039            0.9133356
## LCIP0045 162647    64724            0.6020584
## ECOP0011 384927   179039            0.5348754
## AAIP0093 218572   102759            0.5298620
## LCIP0013 298525   144575            0.5157022
## LCIP0076 193939    94509            0.5126870
## LAIP0007 250777   123132            0.5089980
## BLANK7    12662     6291            0.5031591
## EAIP0032  14378     7437            0.4827514
## ECIP0257 264601   137051            0.4820466
```

## Determine and Plot Error Rates

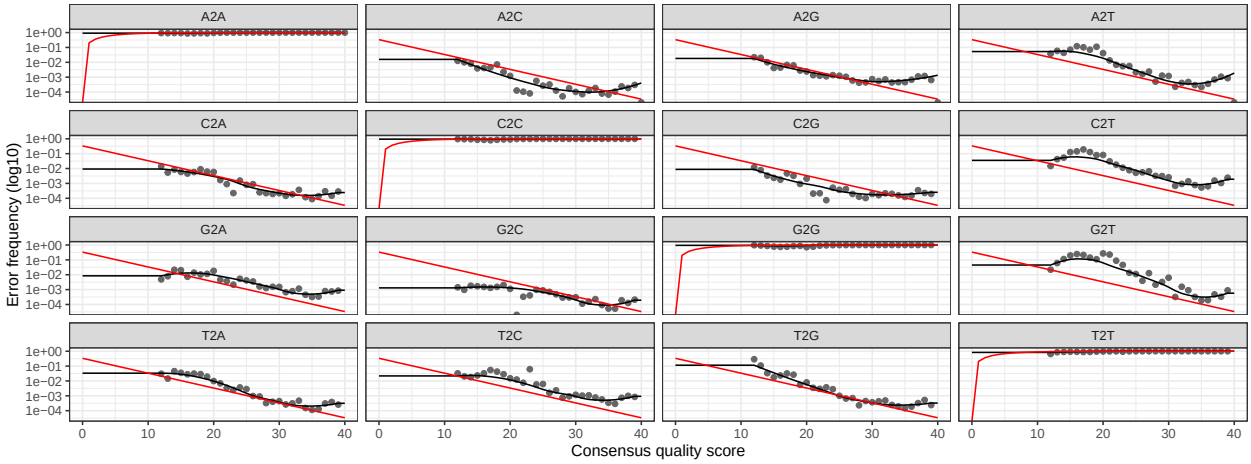
```
error_r1 <- learnErrors(filt_r1, multithread=TRUE)
```

```
## 131659000 total bases in 526636 reads from 5 samples will be used for learning the error rates.
```

```
error_r2 <- learnErrors(filt_r2, multithread=TRUE)
```

```
## 110593560 total bases in 526636 reads from 5 samples will be used for learning the error rates.
```

```
print(plotErrors(error_r1, nominalQ=TRUE))
```




In the plot above the dots are the observed error rates for each consensus quality score. The redline shows the expected error rates based and the black line fit to the observed error rates.

## Sample Inference (Denoise)

```
dada_r1 <- dada(filt_r1, err=error_r1, multithread=TRUE)
```

```
## Sample 1 - 155660 reads in 51239 unique sequences.
## Sample 2 - 17039 reads in 6400 unique sequences.
## Sample 3 - 102759 reads in 17430 unique sequences.
## Sample 4 - 119207 reads in 28553 unique sequences.
## Sample 5 - 131971 reads in 38304 unique sequences.
## Sample 6 - 279408 reads in 69943 unique sequences.
## Sample 7 - 164982 reads in 40076 unique sequences.
## Sample 8 - 197292 reads in 49205 unique sequences.
## Sample 9 - 133398 reads in 37337 unique sequences.
## Sample 10 - 155264 reads in 39651 unique sequences.
## Sample 11 - 140748 reads in 39646 unique sequences.
## Sample 12 - 162560 reads in 34669 unique sequences.
## Sample 13 - 159666 reads in 41975 unique sequences.
## Sample 14 - 152738 reads in 21935 unique sequences.
## Sample 15 - 148259 reads in 26987 unique sequences.
## Sample 16 - 5357 reads in 1804 unique sequences.
## Sample 17 - 7217 reads in 2246 unique sequences.
## Sample 18 - 5445 reads in 1675 unique sequences.
## Sample 19 - 8575 reads in 2190 unique sequences.
## Sample 20 - 12788 reads in 3050 unique sequences.
```

```
## Sample 21 - 4334 reads in 1291 unique sequences.
## Sample 22 - 6291 reads in 1631 unique sequences.
## Sample 23 - 4122 reads in 1320 unique sequences.
## Sample 24 - 115953 reads in 37919 unique sequences.
## Sample 25 - 148092 reads in 43115 unique sequences.
## Sample 26 - 182454 reads in 47996 unique sequences.
## Sample 27 - 7437 reads in 2211 unique sequences.
## Sample 28 - 202524 reads in 60291 unique sequences.
## Sample 29 - 199124 reads in 53887 unique sequences.
## Sample 30 - 158487 reads in 53787 unique sequences.
## Sample 31 - 175309 reads in 53750 unique sequences.
## Sample 32 - 115618 reads in 35324 unique sequences.
## Sample 33 - 138525 reads in 31942 unique sequences.
## Sample 34 - 133484 reads in 38691 unique sequences.
## Sample 35 - 158220 reads in 44830 unique sequences.
## Sample 36 - 149480 reads in 45251 unique sequences.
## Sample 37 - 157621 reads in 47484 unique sequences.
## Sample 38 - 90733 reads in 32864 unique sequences.
## Sample 39 - 196379 reads in 53717 unique sequences.
## Sample 40 - 189327 reads in 40640 unique sequences.
## Sample 41 - 159060 reads in 39702 unique sequences.
## Sample 42 - 132942 reads in 42469 unique sequences.
## Sample 43 - 131743 reads in 38623 unique sequences.
## Sample 44 - 145789 reads in 46470 unique sequences.
## Sample 45 - 133436 reads in 36967 unique sequences.
## Sample 46 - 13202 reads in 4198 unique sequences.
## Sample 47 - 141465 reads in 47069 unique sequences.
## Sample 48 - 113745 reads in 29595 unique sequences.
## Sample 49 - 98910 reads in 34939 unique sequences.
## Sample 50 - 149866 reads in 59138 unique sequences.
## Sample 51 - 169408 reads in 55324 unique sequences.
## Sample 52 - 137921 reads in 35857 unique sequences.
## Sample 53 - 121228 reads in 39708 unique sequences.
## Sample 54 - 108650 reads in 35717 unique sequences.
## Sample 55 - 88758 reads in 31007 unique sequences.
## Sample 56 - 139977 reads in 29441 unique sequences.
## Sample 57 - 181547 reads in 54774 unique sequences.
## Sample 58 - 157745 reads in 41676 unique sequences.
## Sample 59 - 119525 reads in 37464 unique sequences.
## Sample 60 - 175188 reads in 52190 unique sequences.
## Sample 61 - 123971 reads in 43052 unique sequences.
## Sample 62 - 110005 reads in 36660 unique sequences.
## Sample 63 - 109315 reads in 40959 unique sequences.
## Sample 64 - 127193 reads in 45010 unique sequences.
## Sample 65 - 191249 reads in 42181 unique sequences.
## Sample 66 - 108214 reads in 33432 unique sequences.
## Sample 67 - 83023 reads in 29447 unique sequences.
## Sample 68 - 128757 reads in 41639 unique sequences.
## Sample 69 - 121813 reads in 38873 unique sequences.
## Sample 70 - 112792 reads in 38107 unique sequences.
## Sample 71 - 186013 reads in 49896 unique sequences.
## Sample 72 - 180985 reads in 51170 unique sequences.
## Sample 73 - 132051 reads in 38605 unique sequences.
## Sample 74 - 125121 reads in 44123 unique sequences.
```

```
## Sample 75 - 244462 reads in 70122 unique sequences.
## Sample 76 - 211721 reads in 55997 unique sequences.
## Sample 77 - 183545 reads in 52308 unique sequences.
## Sample 78 - 213749 reads in 52253 unique sequences.
## Sample 79 - 183542 reads in 49084 unique sequences.
## Sample 80 - 173196 reads in 48249 unique sequences.
## Sample 81 - 149162 reads in 46873 unique sequences.
## Sample 82 - 152144 reads in 34519 unique sequences.
## Sample 83 - 150804 reads in 46772 unique sequences.
## Sample 84 - 137051 reads in 37969 unique sequences.
## Sample 85 - 186536 reads in 46913 unique sequences.
## Sample 86 - 191379 reads in 50638 unique sequences.
## Sample 87 - 205827 reads in 53517 unique sequences.
## Sample 88 - 179039 reads in 48359 unique sequences.
## Sample 89 - 230864 reads in 58250 unique sequences.
## Sample 90 - 236965 reads in 64291 unique sequences.
## Sample 91 - 155508 reads in 43234 unique sequences.
## Sample 92 - 102747 reads in 28297 unique sequences.
## Sample 93 - 88543 reads in 24452 unique sequences.
## Sample 94 - 93914 reads in 28426 unique sequences.
## Sample 95 - 79109 reads in 21837 unique sequences.
## Sample 96 - 116071 reads in 18297 unique sequences.
## Sample 97 - 149965 reads in 29786 unique sequences.
## Sample 98 - 123132 reads in 40145 unique sequences.
## Sample 99 - 123013 reads in 34357 unique sequences.
## Sample 100 - 103398 reads in 26858 unique sequences.
## Sample 101 - 63005 reads in 16621 unique sequences.
## Sample 102 - 130687 reads in 22128 unique sequences.
## Sample 103 - 144575 reads in 39200 unique sequences.
## Sample 104 - 64724 reads in 17529 unique sequences.
## Sample 105 - 159287 reads in 48527 unique sequences.
## Sample 106 - 153837 reads in 39917 unique sequences.
## Sample 107 - 157789 reads in 39572 unique sequences.
## Sample 108 - 146559 reads in 41267 unique sequences.
## Sample 109 - 94509 reads in 27142 unique sequences.
## Sample 110 - 236042 reads in 50991 unique sequences.
## Sample 111 - 146723 reads in 36600 unique sequences.
```

```
dada_r2 <- dada(filt_r2, err=error_r2, multithread=TRUE)
```

```
## Sample 1 - 155660 reads in 84733 unique sequences.
## Sample 2 - 17039 reads in 9005 unique sequences.
## Sample 3 - 102759 reads in 27152 unique sequences.
## Sample 4 - 119207 reads in 44942 unique sequences.
## Sample 5 - 131971 reads in 68960 unique sequences.
## Sample 6 - 279408 reads in 124412 unique sequences.
## Sample 7 - 164982 reads in 60746 unique sequences.
## Sample 8 - 197292 reads in 80389 unique sequences.
## Sample 9 - 133398 reads in 60908 unique sequences.
## Sample 10 - 155264 reads in 62773 unique sequences.
## Sample 11 - 140748 reads in 59341 unique sequences.
## Sample 12 - 162560 reads in 57110 unique sequences.
## Sample 13 - 159666 reads in 66026 unique sequences.
## Sample 14 - 152738 reads in 31949 unique sequences.
## Sample 15 - 148259 reads in 55645 unique sequences.
```

```
## Sample 16 - 5357 reads in 2809 unique sequences.
## Sample 17 - 7217 reads in 2927 unique sequences.
## Sample 18 - 5445 reads in 2401 unique sequences.
## Sample 19 - 8575 reads in 3605 unique sequences.
## Sample 20 - 12788 reads in 4839 unique sequences.
## Sample 21 - 4334 reads in 1988 unique sequences.
## Sample 22 - 6291 reads in 3199 unique sequences.
## Sample 23 - 4122 reads in 2024 unique sequences.
## Sample 24 - 115953 reads in 54257 unique sequences.
## Sample 25 - 148092 reads in 57189 unique sequences.
## Sample 26 - 182454 reads in 75064 unique sequences.
## Sample 27 - 7437 reads in 3672 unique sequences.
## Sample 28 - 202524 reads in 96418 unique sequences.
## Sample 29 - 199124 reads in 74554 unique sequences.
## Sample 30 - 158487 reads in 80849 unique sequences.
## Sample 31 - 175309 reads in 82720 unique sequences.
## Sample 32 - 115618 reads in 49278 unique sequences.
## Sample 33 - 138525 reads in 46376 unique sequences.
## Sample 34 - 133484 reads in 66312 unique sequences.
## Sample 35 - 158220 reads in 70662 unique sequences.
## Sample 36 - 149480 reads in 78640 unique sequences.
## Sample 37 - 157621 reads in 66446 unique sequences.
## Sample 38 - 90733 reads in 49807 unique sequences.
## Sample 39 - 196379 reads in 109688 unique sequences.
## Sample 40 - 189327 reads in 80730 unique sequences.
## Sample 41 - 159060 reads in 81630 unique sequences.
## Sample 42 - 132942 reads in 83444 unique sequences.
## Sample 43 - 131743 reads in 81379 unique sequences.
## Sample 44 - 145789 reads in 86309 unique sequences.
## Sample 45 - 133436 reads in 70165 unique sequences.
## Sample 46 - 13202 reads in 8266 unique sequences.
## Sample 47 - 141465 reads in 53889 unique sequences.
## Sample 48 - 113745 reads in 30131 unique sequences.
## Sample 49 - 98910 reads in 59805 unique sequences.
## Sample 50 - 149866 reads in 142533 unique sequences.
## Sample 51 - 169408 reads in 62272 unique sequences.
## Sample 52 - 137921 reads in 41014 unique sequences.
## Sample 53 - 121228 reads in 42774 unique sequences.
## Sample 54 - 108650 reads in 34650 unique sequences.
## Sample 55 - 88758 reads in 33686 unique sequences.
## Sample 56 - 139977 reads in 44436 unique sequences.
## Sample 57 - 181547 reads in 54761 unique sequences.
## Sample 58 - 157745 reads in 63525 unique sequences.
## Sample 59 - 119525 reads in 41042 unique sequences.
## Sample 60 - 175188 reads in 58219 unique sequences.
## Sample 61 - 123971 reads in 46686 unique sequences.
## Sample 62 - 110005 reads in 46848 unique sequences.
## Sample 63 - 109315 reads in 52102 unique sequences.
## Sample 64 - 127193 reads in 56363 unique sequences.
## Sample 65 - 191249 reads in 88345 unique sequences.
## Sample 66 - 108214 reads in 47578 unique sequences.
## Sample 67 - 83023 reads in 33497 unique sequences.
## Sample 68 - 128757 reads in 52035 unique sequences.
## Sample 69 - 121813 reads in 43841 unique sequences.
```

```
## Sample 70 - 112792 reads in 46482 unique sequences.
## Sample 71 - 186013 reads in 55705 unique sequences.
## Sample 72 - 180985 reads in 61421 unique sequences.
## Sample 73 - 132051 reads in 41402 unique sequences.
## Sample 74 - 125121 reads in 52561 unique sequences.
## Sample 75 - 244462 reads in 70861 unique sequences.
## Sample 76 - 211721 reads in 62206 unique sequences.
## Sample 77 - 183545 reads in 71050 unique sequences.
## Sample 78 - 213749 reads in 79398 unique sequences.
## Sample 79 - 183542 reads in 74216 unique sequences.
## Sample 80 - 173196 reads in 76982 unique sequences.
## Sample 81 - 149162 reads in 63874 unique sequences.
## Sample 82 - 152144 reads in 46855 unique sequences.
## Sample 83 - 150804 reads in 65402 unique sequences.
## Sample 84 - 137051 reads in 58157 unique sequences.
## Sample 85 - 186536 reads in 60616 unique sequences.
## Sample 86 - 191379 reads in 69030 unique sequences.
## Sample 87 - 205827 reads in 70678 unique sequences.
## Sample 88 - 179039 reads in 67384 unique sequences.
## Sample 89 - 230864 reads in 63271 unique sequences.
## Sample 90 - 236965 reads in 91692 unique sequences.
## Sample 91 - 155508 reads in 65123 unique sequences.
## Sample 92 - 102747 reads in 47764 unique sequences.
## Sample 93 - 88543 reads in 42362 unique sequences.
## Sample 94 - 93914 reads in 44714 unique sequences.
## Sample 95 - 79109 reads in 34064 unique sequences.
## Sample 96 - 116071 reads in 28002 unique sequences.
## Sample 97 - 149965 reads in 43758 unique sequences.
## Sample 98 - 123132 reads in 51571 unique sequences.
## Sample 99 - 123013 reads in 54683 unique sequences.
## Sample 100 - 103398 reads in 40366 unique sequences.
## Sample 101 - 63005 reads in 21232 unique sequences.
## Sample 102 - 130687 reads in 25898 unique sequences.
## Sample 103 - 144575 reads in 56004 unique sequences.
## Sample 104 - 64724 reads in 31825 unique sequences.
## Sample 105 - 159287 reads in 78244 unique sequences.
## Sample 106 - 153837 reads in 83859 unique sequences.
## Sample 107 - 157789 reads in 72555 unique sequences.
## Sample 108 - 146559 reads in 69258 unique sequences.
## Sample 109 - 94509 reads in 47622 unique sequences.
## Sample 110 - 236042 reads in 76514 unique sequences.
## Sample 111 - 146723 reads in 36092 unique sequences.
```

### Merge Reads and Construct Amplicon Sequence Variant (ASV) Table

```
merged_reads <- mergePairs(dada_r1, filt_r1, dada_r2, filt_r2, verbose=TRUE)
```

```
## 109639 paired-reads (in 368 unique pairings) successfully merged out of 152448 (in 1525 pairings) inp
## 13460 paired-reads (in 138 unique pairings) successfully merged out of 16420 (in 310 pairings) input
## 72766 paired-reads (in 187 unique pairings) successfully merged out of 101950 (in 385 pairings) input
## 98417 paired-reads (in 275 unique pairings) successfully merged out of 118091 (in 706 pairings) input
```

```
## 106655 paired-reads (in 441 unique pairings) successfully merged out of 129535 (in 1549 pairings) inp
## 198688 paired-reads (in 595 unique pairings) successfully merged out of 275429 (in 2301 pairings) inp
## 130317 paired-reads (in 406 unique pairings) successfully merged out of 163191 (in 1226 pairings) inp
## 165687 paired-reads (in 453 unique pairings) successfully merged out of 194548 (in 1130 pairings) inp
## 110070 paired-reads (in 380 unique pairings) successfully merged out of 131156 (in 938 pairings) inpu
## 128154 paired-reads (in 401 unique pairings) successfully merged out of 153086 (in 1487 pairings) inp
## 110217 paired-reads (in 424 unique pairings) successfully merged out of 138251 (in 1563 pairings) inp
## 110491 paired-reads (in 416 unique pairings) successfully merged out of 159946 (in 1149 pairings) inp
## 127589 paired-reads (in 420 unique pairings) successfully merged out of 156668 (in 1296 pairings) inp
## 91988 paired-reads (in 149 unique pairings) successfully merged out of 152101 (in 645 pairings) input
## 99314 paired-reads (in 297 unique pairings) successfully merged out of 147221 (in 757 pairings) input
## 3564 paired-reads (in 17 unique pairings) successfully merged out of 5207 (in 38 pairings) input.
## 5205 paired-reads (in 20 unique pairings) successfully merged out of 7058 (in 47 pairings) input.
## 3170 paired-reads (in 13 unique pairings) successfully merged out of 5295 (in 27 pairings) input.
## 5190 paired-reads (in 20 unique pairings) successfully merged out of 8400 (in 36 pairings) input.
## 7915 paired-reads (in 21 unique pairings) successfully merged out of 12589 (in 49 pairings) input.
## 3251 paired-reads (in 18 unique pairings) successfully merged out of 4185 (in 27 pairings) input.
## 4318 paired-reads (in 20 unique pairings) successfully merged out of 6155 (in 26 pairings) input.
## 3202 paired-reads (in 24 unique pairings) successfully merged out of 3998 (in 30 pairings) input.
## 82363 paired-reads (in 311 unique pairings) successfully merged out of 113685 (in 1125 pairings) inpu
## 107689 paired-reads (in 408 unique pairings) successfully merged out of 145025 (in 1491 pairings) inp
## 122664 paired-reads (in 464 unique pairings) successfully merged out of 179859 (in 2269 pairings) inp
## 4381 paired-reads (in 17 unique pairings) successfully merged out of 7145 (in 37 pairings) input.
## 148260 paired-reads (in 624 unique pairings) successfully merged out of 198674 (in 2204 pairings) inp
## 142612 paired-reads (in 349 unique pairings) successfully merged out of 196863 (in 928 pairings) inpu
## 103608 paired-reads (in 538 unique pairings) successfully merged out of 154738 (in 2353 pairings) inp
## 113952 paired-reads (in 478 unique pairings) successfully merged out of 172011 (in 2214 pairings) inp
## 86141 paired-reads (in 370 unique pairings) successfully merged out of 112306 (in 1289 pairings) inpu
## 85113 paired-reads (in 314 unique pairings) successfully merged out of 136988 (in 1003 pairings) inpu
## 92027 paired-reads (in 425 unique pairings) successfully merged out of 130711 (in 2274 pairings) inpu
## 116347 paired-reads (in 412 unique pairings) successfully merged out of 155067 (in 1426 pairings) inp
## 107294 paired-reads (in 410 unique pairings) successfully merged out of 145765 (in 2002 pairings) inp
## 111798 paired-reads (in 517 unique pairings) successfully merged out of 153594 (in 1677 pairings) inp
## 61052 paired-reads (in 342 unique pairings) successfully merged out of 87931 (in 1375 pairings) input
## 129923 paired-reads (in 474 unique pairings) successfully merged out of 191838 (in 1864 pairings) inp
## 162754 paired-reads (in 404 unique pairings) successfully merged out of 187129 (in 1092 pairings) inp
```

```
## 117497 paired-reads (in 377 unique pairings) successfully merged out of 156560 (in 1249 pairings) inp
## 101150 paired-reads (in 348 unique pairings) successfully merged out of 129623 (in 1405 pairings) inp
## 91745 paired-reads (in 278 unique pairings) successfully merged out of 129244 (in 1039 pairings) inpu
## 110643 paired-reads (in 434 unique pairings) successfully merged out of 140925 (in 1578 pairings) inp
## 87070 paired-reads (in 326 unique pairings) successfully merged out of 129951 (in 1265 pairings) inpu
## 10872 paired-reads (in 28 unique pairings) successfully merged out of 13002 (in 53 pairings) input.
## 111270 paired-reads (in 521 unique pairings) successfully merged out of 138190 (in 1683 pairings) inp
## 79641 paired-reads (in 327 unique pairings) successfully merged out of 112140 (in 951 pairings) input
## 49793 paired-reads (in 290 unique pairings) successfully merged out of 96436 (in 1289 pairings) input
## 59231 paired-reads (in 83 unique pairings) successfully merged out of 123514 (in 688 pairings) input
## 98435 paired-reads (in 540 unique pairings) successfully merged out of 165534 (in 2906 pairings) inpu
## 100108 paired-reads (in 319 unique pairings) successfully merged out of 136513 (in 1120 pairings) inp
## 99944 paired-reads (in 534 unique pairings) successfully merged out of 118530 (in 1668 pairings) inpu
## 77432 paired-reads (in 462 unique pairings) successfully merged out of 106066 (in 1438 pairings) inpu
## 62260 paired-reads (in 401 unique pairings) successfully merged out of 86383 (in 1601 pairings) input
## 100206 paired-reads (in 334 unique pairings) successfully merged out of 137905 (in 1090 pairings) inp
## 143172 paired-reads (in 597 unique pairings) successfully merged out of 178603 (in 1887 pairings) inp
## 114946 paired-reads (in 457 unique pairings) successfully merged out of 155402 (in 1511 pairings) inp
## 83754 paired-reads (in 401 unique pairings) successfully merged out of 116981 (in 1732 pairings) inpu
## 94330 paired-reads (in 455 unique pairings) successfully merged out of 172544 (in 1709 pairings) inpu
## 96348 paired-reads (in 493 unique pairings) successfully merged out of 120964 (in 1418 pairings) inpu
## 64145 paired-reads (in 438 unique pairings) successfully merged out of 107730 (in 1896 pairings) inpu
## 77851 paired-reads (in 572 unique pairings) successfully merged out of 106032 (in 2185 pairings) inpu
## 80412 paired-reads (in 412 unique pairings) successfully merged out of 124906 (in 2073 pairings) inpu
## 132201 paired-reads (in 479 unique pairings) successfully merged out of 188561 (in 1669 pairings) inp
## 74264 paired-reads (in 303 unique pairings) successfully merged out of 106849 (in 1321 pairings) inpu
## 51370 paired-reads (in 228 unique pairings) successfully merged out of 80508 (in 1149 pairings) input
## 98478 paired-reads (in 536 unique pairings) successfully merged out of 125916 (in 1832 pairings) inpu
## 93877 paired-reads (in 417 unique pairings) successfully merged out of 119646 (in 1475 pairings) inpu
## 69447 paired-reads (in 401 unique pairings) successfully merged out of 110295 (in 2050 pairings) inpu
## 139444 paired-reads (in 572 unique pairings) successfully merged out of 182980 (in 2011 pairings) inp
## 130006 paired-reads (in 667 unique pairings) successfully merged out of 178132 (in 2835 pairings) inp
## 111634 paired-reads (in 353 unique pairings) successfully merged out of 130227 (in 1275 pairings) inp
## 105969 paired-reads (in 508 unique pairings) successfully merged out of 121471 (in 1743 pairings) inp
## 174072 paired-reads (in 775 unique pairings) successfully merged out of 240896 (in 2387 pairings) inp
## 155489 paired-reads (in 611 unique pairings) successfully merged out of 208590 (in 2172 pairings) inp
```

```
## 146455 paired-reads (in 633 unique pairings) successfully merged out of 180821 (in 1732 pairings) inp
## 175840 paired-reads (in 619 unique pairings) successfully merged out of 209336 (in 1893 pairings) inp
## 149465 paired-reads (in 590 unique pairings) successfully merged out of 180383 (in 1664 pairings) inp
## 130916 paired-reads (in 485 unique pairings) successfully merged out of 169565 (in 1744 pairings) inp
## 109177 paired-reads (in 524 unique pairings) successfully merged out of 145047 (in 1519 pairings) inp
## 121764 paired-reads (in 373 unique pairings) successfully merged out of 150538 (in 1178 pairings) inp
## 119697 paired-reads (in 537 unique pairings) successfully merged out of 148243 (in 2069 pairings) inp
## 109952 paired-reads (in 425 unique pairings) successfully merged out of 134741 (in 1152 pairings) inp
## 166222 paired-reads (in 578 unique pairings) successfully merged out of 184237 (in 1437 pairings) inp
## 153340 paired-reads (in 601 unique pairings) successfully merged out of 188232 (in 1645 pairings) inp
## 165128 paired-reads (in 605 unique pairings) successfully merged out of 202716 (in 1830 pairings) inp
## 132842 paired-reads (in 538 unique pairings) successfully merged out of 176417 (in 1785 pairings) inp
## 190988 paired-reads (in 698 unique pairings) successfully merged out of 227326 (in 1717 pairings) inp
## 180327 paired-reads (in 674 unique pairings) successfully merged out of 233269 (in 2516 pairings) inp
## 109114 paired-reads (in 467 unique pairings) successfully merged out of 153333 (in 1781 pairings) inp
## 68044 paired-reads (in 283 unique pairings) successfully merged out of 100946 (in 904 pairings) input
## 67893 paired-reads (in 298 unique pairings) successfully merged out of 86869 (in 1024 pairings) input
## 73835 paired-reads (in 356 unique pairings) successfully merged out of 92178 (in 967 pairings) input
## 63703 paired-reads (in 231 unique pairings) successfully merged out of 77753 (in 473 pairings) input
## 94746 paired-reads (in 78 unique pairings) successfully merged out of 115421 (in 262 pairings) input
## 117341 paired-reads (in 295 unique pairings) successfully merged out of 148385 (in 738 pairings) inpu
## 82781 paired-reads (in 376 unique pairings) successfully merged out of 121614 (in 897 pairings) input
## 97570 paired-reads (in 408 unique pairings) successfully merged out of 120451 (in 961 pairings) input
## 89221 paired-reads (in 404 unique pairings) successfully merged out of 102014 (in 921 pairings) input
## 53294 paired-reads (in 179 unique pairings) successfully merged out of 61660 (in 302 pairings) input
## 110852 paired-reads (in 125 unique pairings) successfully merged out of 129955 (in 372 pairings) inpu
## 102543 paired-reads (in 409 unique pairings) successfully merged out of 142630 (in 845 pairings) inpu
## 47494 paired-reads (in 177 unique pairings) successfully merged out of 63229 (in 542 pairings) input
## 115088 paired-reads (in 520 unique pairings) successfully merged out of 156148 (in 1408 pairings) inp
## 109213 paired-reads (in 318 unique pairings) successfully merged out of 150020 (in 1057 pairings) inp
## 130508 paired-reads (in 385 unique pairings) successfully merged out of 155303 (in 1231 pairings) inp
## 115773 paired-reads (in 364 unique pairings) successfully merged out of 144674 (in 927 pairings) inpu
## 77112 paired-reads (in 293 unique pairings) successfully merged out of 92926 (in 783 pairings) input
## 189097 paired-reads (in 374 unique pairings) successfully merged out of 233951 (in 1006 pairings) inp
## 106801 paired-reads (in 410 unique pairings) successfully merged out of 145509 (in 1279 pairings) inp
```

```
head(merged_reads[[1]])
```

```
##
## 1              AGAGTTTGATCCTGGCTCAGGACGAACGCTGGCGGCGCGCCTAACACATGCAAGTCGAACGGAGCTGAGAGGAGCTTGCTTTTCT
## 2           AGAGTTTGATCCTGGCTCAGGATGAACGCTGGCGGCATGCCTAATACATGCAAGTCGAACGAGAGGAAGGAAAGCTTGCTTTTCTGAA
## 3           AGAGTTTGATCCTGGCTCAGGATGAACGCTGGCGGCATGCCTAATACATGCAAGTCGAACGAGAGGAAGGAAAGCTTGCTTTTCTGAA
## 4 AGAGTTTGATCCTGGCTCAGGATGAACGCTGGCGGCGTGCTTAACACATGCAAGTCGAACGAAGCACTTTATTTGATTTCCTTCGGGACTGATTATTTT
## 6              AGAGTTTGATCCTGGCTCAGGATGAACGCTAGCGGCAGGCTTAACACATGCAAGTCGAGGGGCAGCATAATGGATAGCAATATCTA
## 7              AGAGTTTGATCCTGGCTCAGGATGAACGCTAGCGGCAGGCTTAACACATGCAAGTCGAGGGGCAACATAATGGATAGCAATATCTA
##   abundance forward reverse nmatch nmismatch nindel prefer accept
## 1      7492       2       3    108         0      0      1   TRUE
## 2      7454       1       1    105         0      0      1   TRUE
## 3      4817       3       2    105         0      0      1   TRUE
## 4      3046       6       5     95         0      0      1   TRUE
## 6      2479       7       4    108         0      0      1   TRUE
## 7      2437       8       4    108         0      0      1   TRUE
```

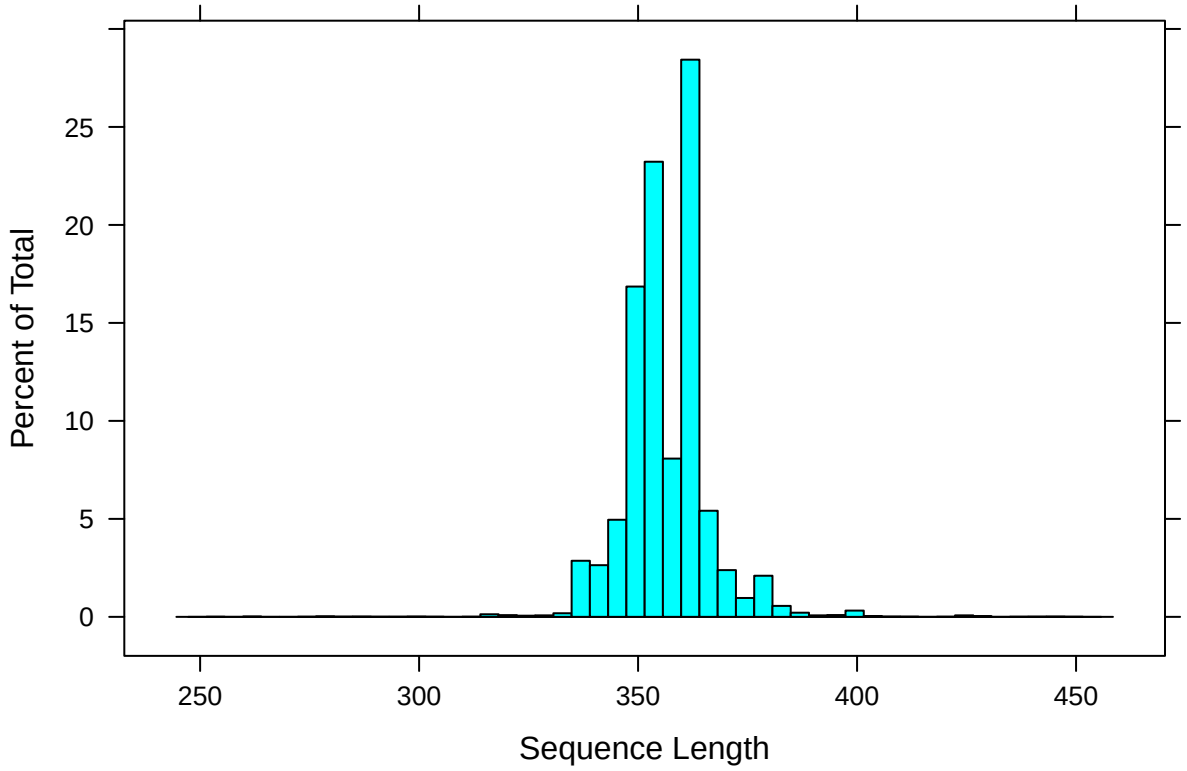
```
sequence_table <- makeSequenceTable(merged_reads)
dim(sequence_table)
```

```
## [1]   111 25025
```

```
table(nchar(getSequences(sequence_table)))
```

```
##
##  255  260  262  263  275  277  278  280  281  286  287  292  296  300  301
##    1    2    1    1    1    3    1    2    2    2    1    1    1    2    1
##  304  313  317  318  319  320  321  322  323  324  325  326  327  328  329
##    1    1   32    1    8    1    2   11    2    4    7    2    8    6    1
##  330  331  332  333  336  337  340  341  342  343  344  345  346  347  348
##    4    9   16   21  712    4   65  280   12  303  138  144  514  444  513
##  349  350  351  352  353  354  355  356  357  358  359  360  361  362  363
## 2079 1030  596 1522 1013 1228 2049 1777   18   31  195  377  772 3163 2140
##  364  365  366  367  368  369  370  371  372  373  374  375  376  377  378
##  663  450   22  590  293   44   86  418   49   52   82   86   21   50  168
##  379  380  381  382  383  385  386  387  388  389  391  392  394  395  396
##  166  141   72   46   22    1   44    2    3    3   16    3    3    6    3
##  397  398  399  400  401  402  403  404  405  406  409  410  413  421  423
##   11   22   47    6    5    1    4    2    3    2    1    1    1    1   19
##  429  430  438  440  442  444  448
##    1    8    1    1    1    3    1
```

```
print(histogram(nchar(getSequences(sequence_table)), nint=50, xlab="Sequence Length"))
```

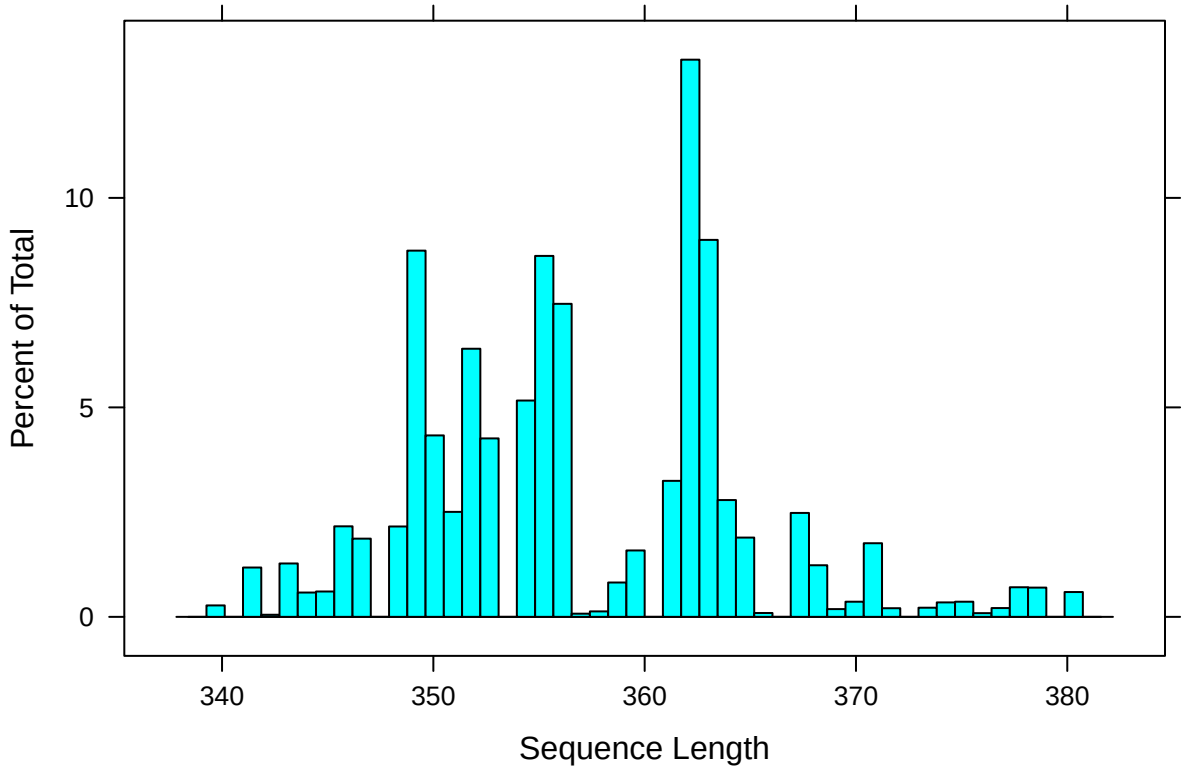



Filter out merged reads that have a length that is not within 20bp of our expected ~360bp. These could be the result of non-specific priming.

```
sequence_table <- sequence_table[,nchar(colnames(sequence_table)) %in% 340:380]
table(nchar(getSequences(sequence_table)))
```

```
##
##  340  341  342  343  344  345  346  347  348  349  350  351  352  353  354
##   65  280   12  303  138  144  514  444  513 2079 1030  596 1522 1013 1228
##  355  356  357  358  359  360  361  362  363  364  365  366  367  368  369
## 2049 1777   18   31  195  377  772 3163 2140  663  450   22  590  293   44
##  370  371  372  373  374  375  376  377  378  379  380
##   86  418   49   52   82   86   21   50  168  166  141
```

```
print(histogram(nchar(getSequences(sequence_table)), nint=50, xlab="Sequence Length"))
```




### Remove Chimeras

```
sequence_table_nochim <- removeBimeraDenovo(
  sequence_table, method="consensus", multithread=TRUE, verbose=TRUE
)
```

```
## Identified 14171 bimeras out of 23784 input sequences.
```

```
dim(sequence_table_nochim)
```

```
## [1]  111 9613
```

```
1 - (sum(sequence_table_nochim) / sum(sequence_table))
```

```
## [1] 0.04209585
```

## Track Reads

```
getN <- function(x) sum(getUniques(x))
track <- as.data.frame(cbind(out, sapply(dada_r1, getN), sapply(dada_r2, getN), sapply(merged_reads, ge
colnames(track) <- c("input", "filtered", "denoisedR1", "denoisedR2", "merged", "nonchim")
rownames(track) <- sample_names
track$percent_filtered_out <- 1 - (track$filtered / track$input)
track$percent_merged <- track$merged / track$filtered
track$percent_nochim <- track$nonchim / track$merged
head(track)
```

```
##           input filtered denoisedR1 denoisedR2 merged nonchim
## AAIP0060 247847   155660     153810     154071 109639  102150
## AAIP0064 196609    17039      16701      16648  13460   11563
## AAIP0093 218572   102759     102384     102243  72766   69665
## AAIP0094 189536   119207     118594     118631  98417   90301
## AAIP0119 208551   131971     130578     130753 106655   97371
## AAIP0125 460087   279408     277133     277521 198688  143910
##          percent_filtered_out percent_merged percent_nochim
## AAIP0060            0.3719512      0.7043492      0.9316940
## AAIP0064            0.9133356      0.7899525      0.8590639
## AAIP0093            0.5298620      0.7081229      0.9573839
## AAIP0094            0.3710588      0.8255975      0.9175346
## AAIP0119            0.3672003      0.8081700      0.9129530
## AAIP0125            0.3927062      0.7111035      0.7243014
```

## Assign Taxonomy

The SILVA (v132) DADA2-formatted reference database was used to assign taxonomy (including species) to the ASVs. The SILVA (v132) DADA2 formatted databases are avialable at DADA2 Reference Databases.

```
taxa <- assignTaxonomy(
  sequence_table_nochim,
  paste0(PROJECT_DIR, "/data/reference/silva_nr_v132_train_set.fa.gz"),
  tryRC=TRUE,
  multithread=TRUE
)
taxa <- addSpecies(
  taxa, paste0(PROJECT_DIR, "/data/reference/silva_species_assignment_v132.fa.gz")
)
taxa.print <- taxa
rownames(taxa.print) <- NULL
head(taxa.print)
```

```
##      Kingdom    Phylum           Class                 Order
## [1,] "Bacteria" "Bacteroidetes"  "Bacteroidia"         "Bacteroidales"
## [2,] "Bacteria" "Firmicutes"     "Clostridia"          "Clostridiales"
## [3,] "Bacteria" "Proteobacteria" "Gammaproteobacteria" "Enterobacteriales"
## [4,] "Bacteria" "Proteobacteria" "Gammaproteobacteria" "Enterobacteriales"
## [5,] "Bacteria" "Bacteroidetes"  "Bacteroidia"         "Bacteroidales"
## [6,] "Bacteria" "Firmicutes"     "Bacilli"             "Lactobacillales"
##      Family               Genus                 Species
## [1,] "Bacteroidaceae"     "Bacteroides"         "vulgatus"
## [2,] "Ruminococcaceae"    "Subdoligranulum"     NA
## [3,] "Enterobacteriaceae" "Pseudocitrobacter"   NA
## [4,] "Enterobacteriaceae" "Escherichia/Shigella" NA
## [5,] "Rikenellaceae"      "Alistipes"           "putredinis"
## [6,] "Lactobacillaceae"   "Lactobacillus"       NA
```

## Write Outputs

### ASV FASTA

```
asv_seqs <- colnames(sequence_table_nochim)
asv_headers <- vector(dim(sequence_table_nochim)[2], mode="character")
for (i in 1:dim(sequence_table_nochim)[2]) {
  asv_headers[i] <- paste(">ASV", i, sep="_")
}
asv_fasta <- c(rbind(asv_headers, asv_seqs))
write(asv_fasta, paste0(PROJECT_DIR, "/results/ASVs.fa"))
```

### ASV Counts and Taxonomy

```
asv_tab <- t(sequence_table_nochim)
row.names(asv_tab) <- sub(">", "", asv_headers)
write.table(
  asv_tab,
  paste0(PROJECT_DIR, "/results/ASVs-counts.tsv"),
  sep="\t",
  quote=F,
  col.names=NA
)

asv_tax <- taxa
row.names(asv_tax) <- sub(">", "", asv_headers)
write.table(
  asv_tax,
  paste0(PROJECT_DIR, "/results/ASVs-taxonomy.tsv"),
  sep="\t",
  quote=F,
  col.names=NA
)
```

## Remove Contaminants

Here decontam is used to identify likely contaminants using the included *BLANK* samples.

```
vector_for_decontam <- grepl("BLANK", colnames(asv_tab))
contam_asvs <- isContaminant(t(asv_tab), neg=vector_for_decontam)
table(contam_asvs$contaminant)
```

```
##
## FALSE  TRUE
##  9589    24
```

```
asv_tax[row.names(asv_tax) %in% contam_asvs, ]
```

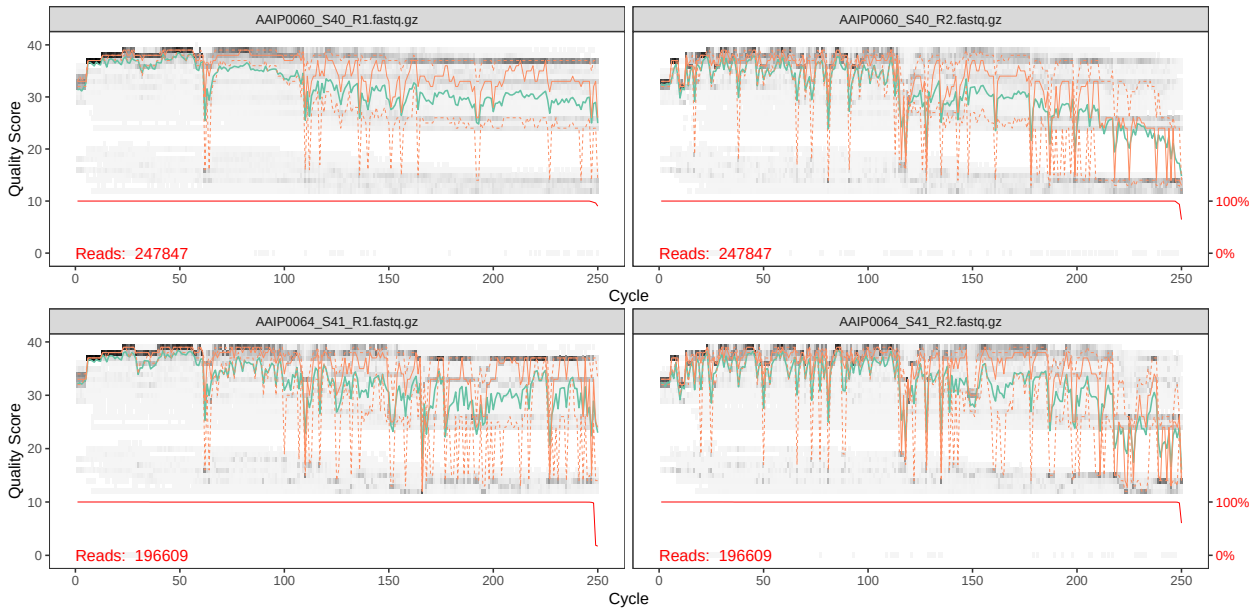
```
##      Kingdom Phylum Class Order Family Genus Species
```

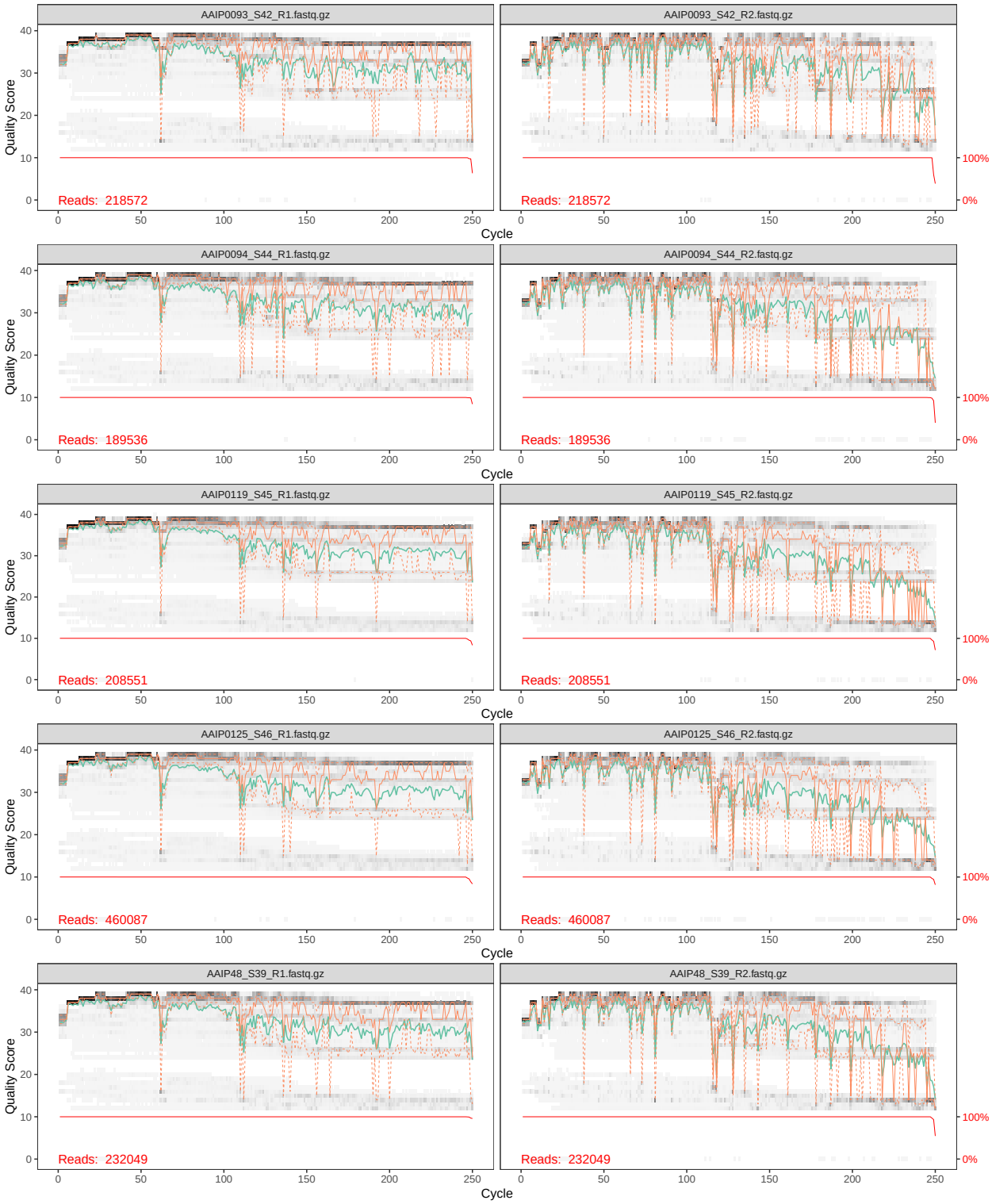
### Write Decontaminated Outputs

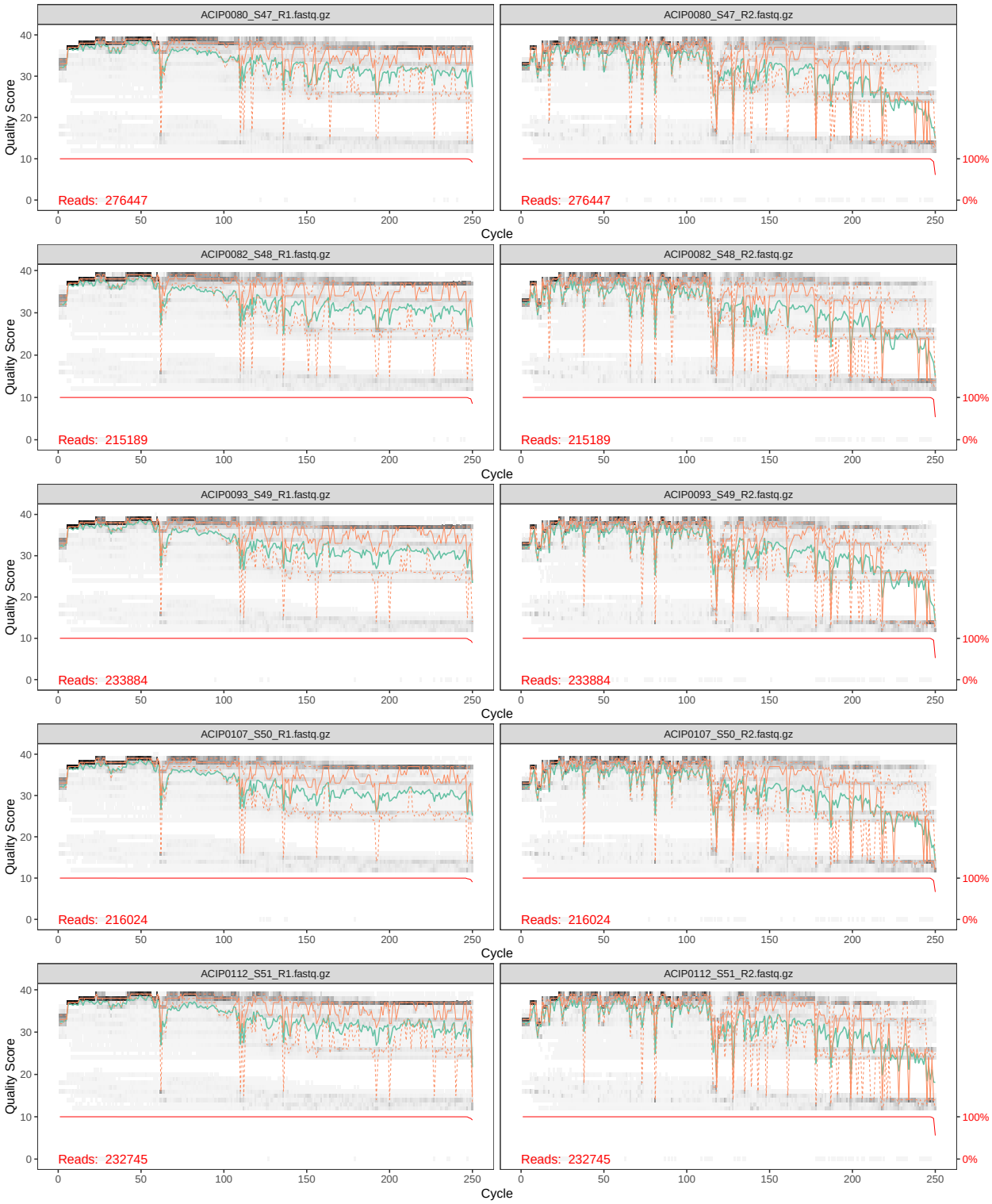
```
contam_indices <- which(asv_fasta %in% paste0(">", contam_asvs))
dont_want <- sort(c(contam_indices, contam_indices + 1))

asv_fasta_no_contam <- asv_fasta[- dont_want]
write(asv_fasta_no_contam, paste0(PROJECT_DIR, "/results/ASVs-decontam.fa"))

asv_tab_no_contam <- asv_tab[!row.names(asv_tab) %in% contam_asvs, ]
write.table(
  asv_tab_no_contam,
  paste0(PROJECT_DIR, "/results/ASVs-counts-decontam.tsv"),
  sep="\t",
  quote=F,
  col.names=NA
)

asv_tax_no_contam <- asv_tax[!row.names(asv_tax) %in% contam_asvs, ]
write.table(
  asv_tax_no_contam,
  paste0(PROJECT_DIR, "/results/ASVs-taxonomy-decontam.tsv"),
  sep="\t",
  quote=F,
  col.names=NA
)
```

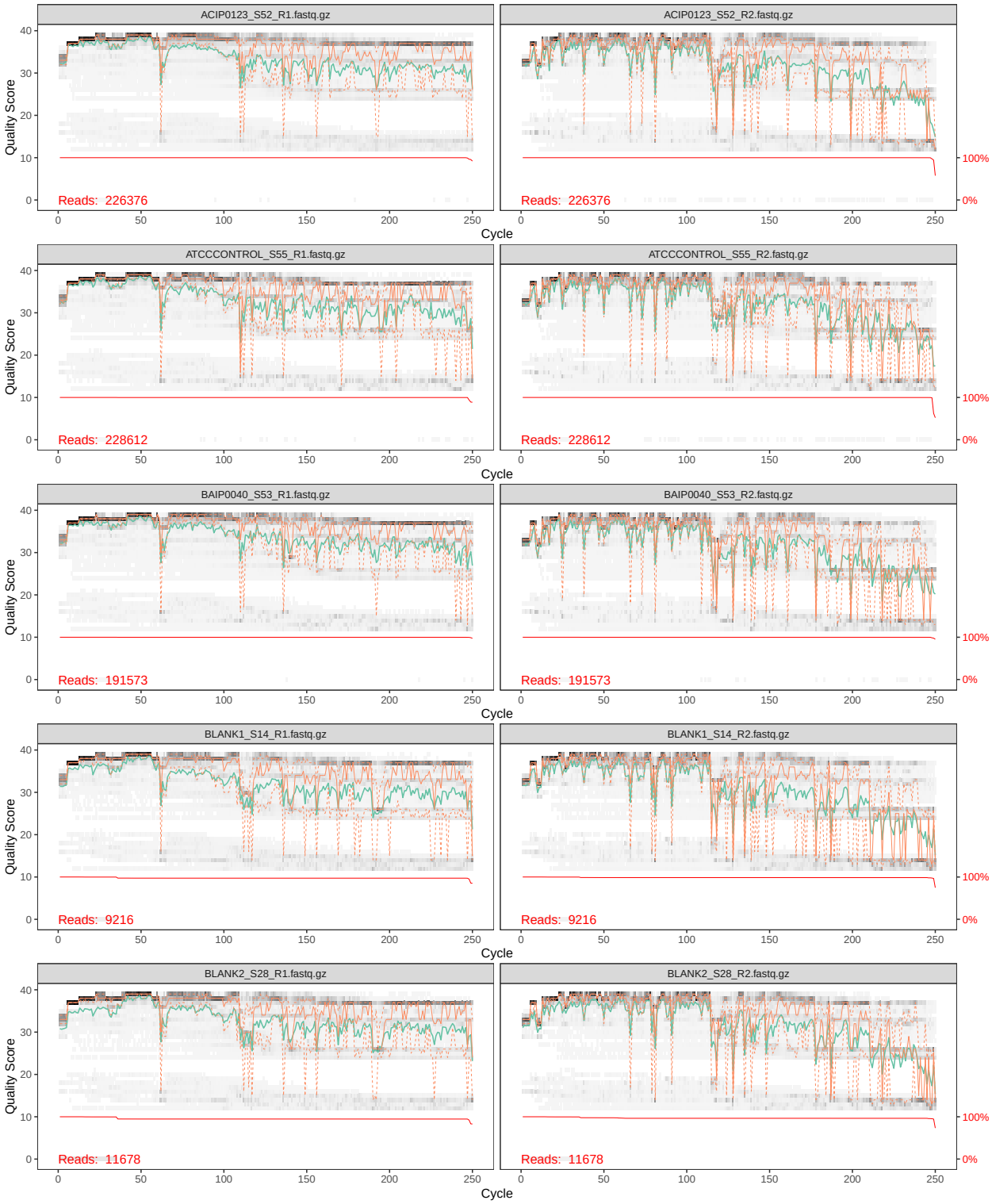
## Plot Sample FASTQ Quality Profiles (Full Set)

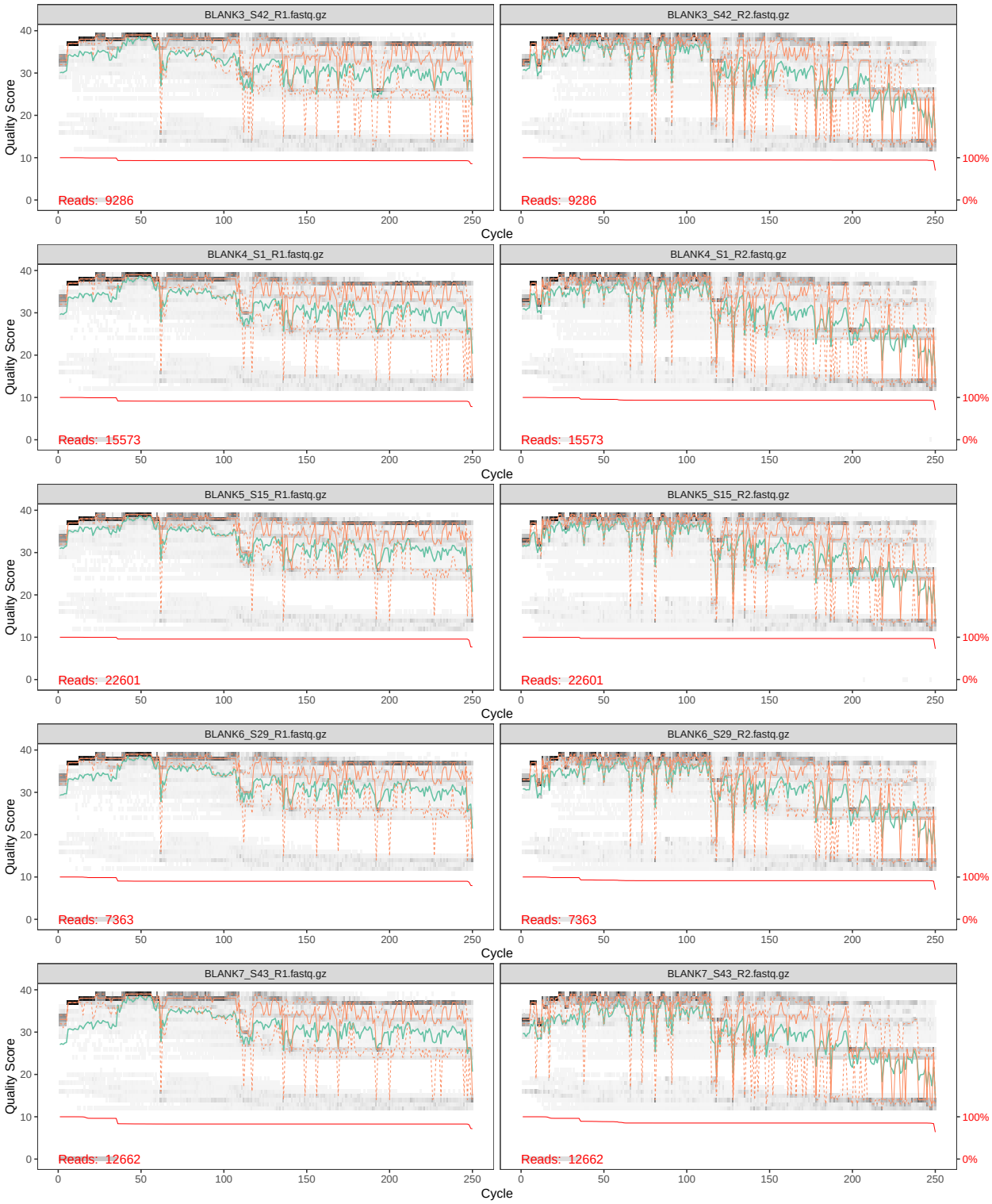
```
for (i in 1:length(fastq_r1)) {
  print(plotQualityProfile(c(fastq_r1[i], fastq_r2[i])))
}
```

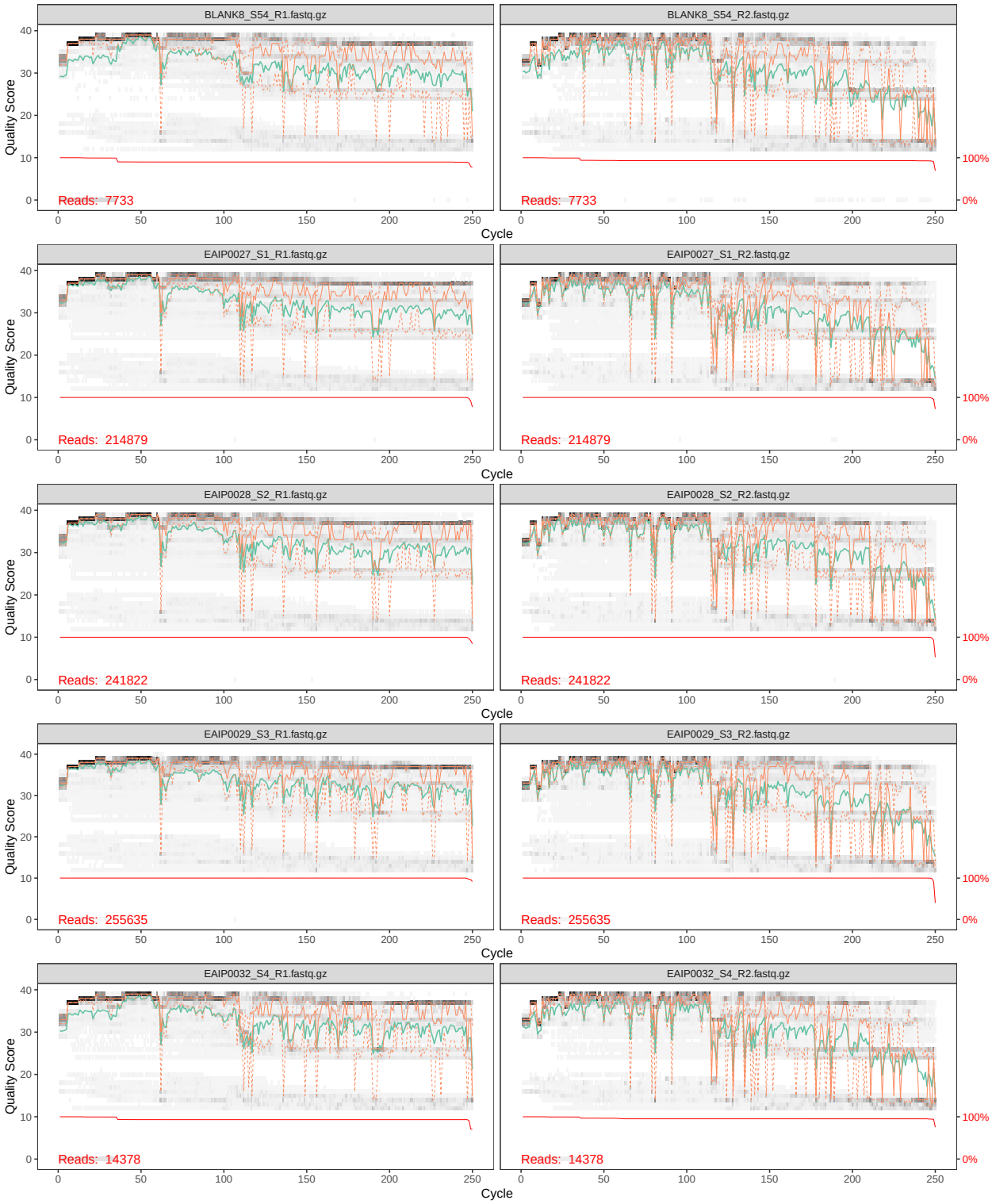


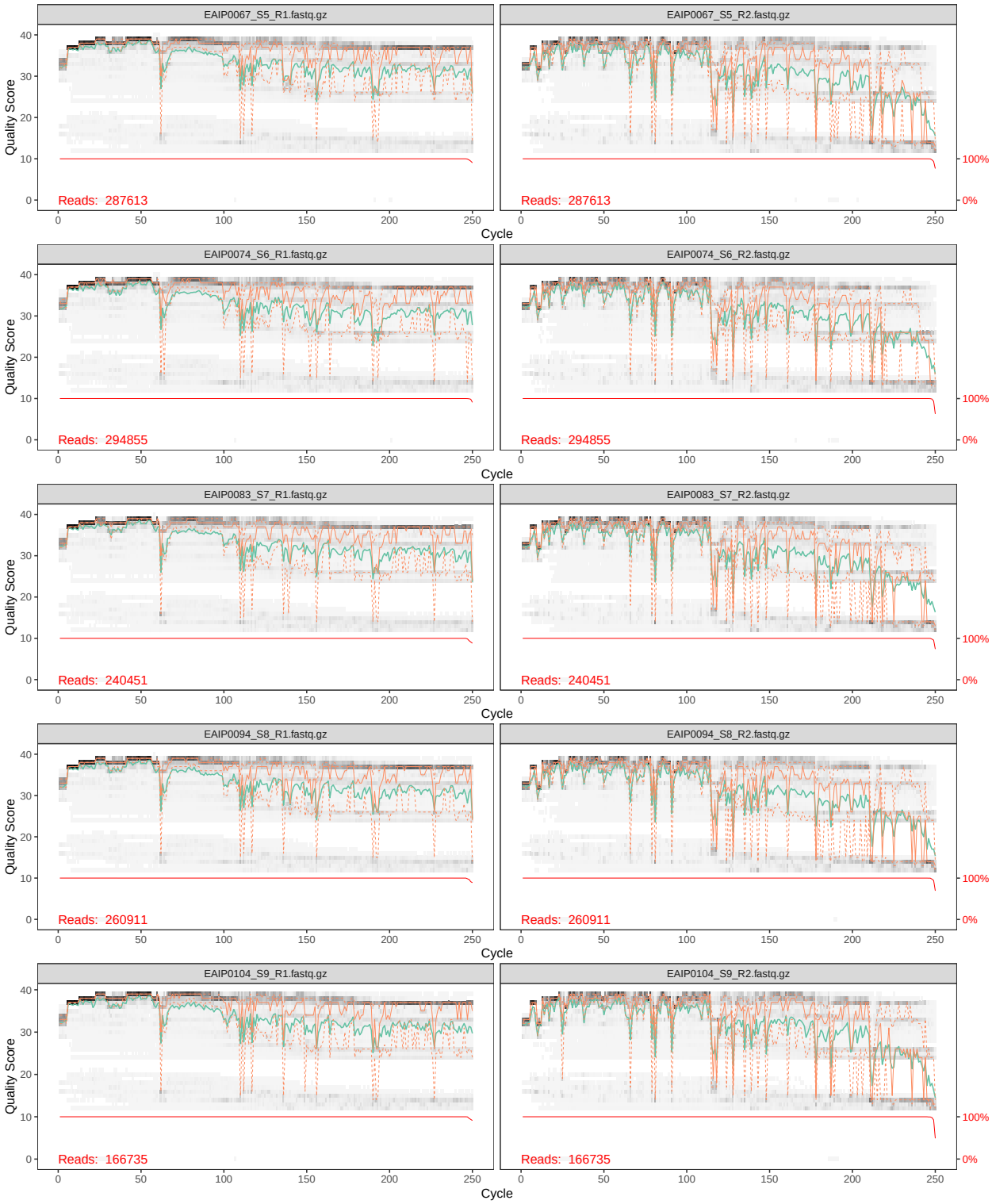

AAIP0093_S42_R1.fastq.gz
AAIP0093_S42_R2.fastq.gz
AAIP0094_S44_R1.fastq.gz
AAIP0094_S44_R2.fastq.gz
AAIP0119_S45_R1.fastq.gz
AAIP0119_S45_R2.fastq.gz
AAIP0125_S46_R1.fastq.gz
AAIP0125_S46_R2.fastq.gz
AAIP48_S39_R1.fastq.gz
AAIP48_S39_R2.fastq.gz
Quality Score
Cycle
Reads: 218572
Reads: 189536
Reads: 208551
Reads: 460087
Reads: 232049
100%
0%

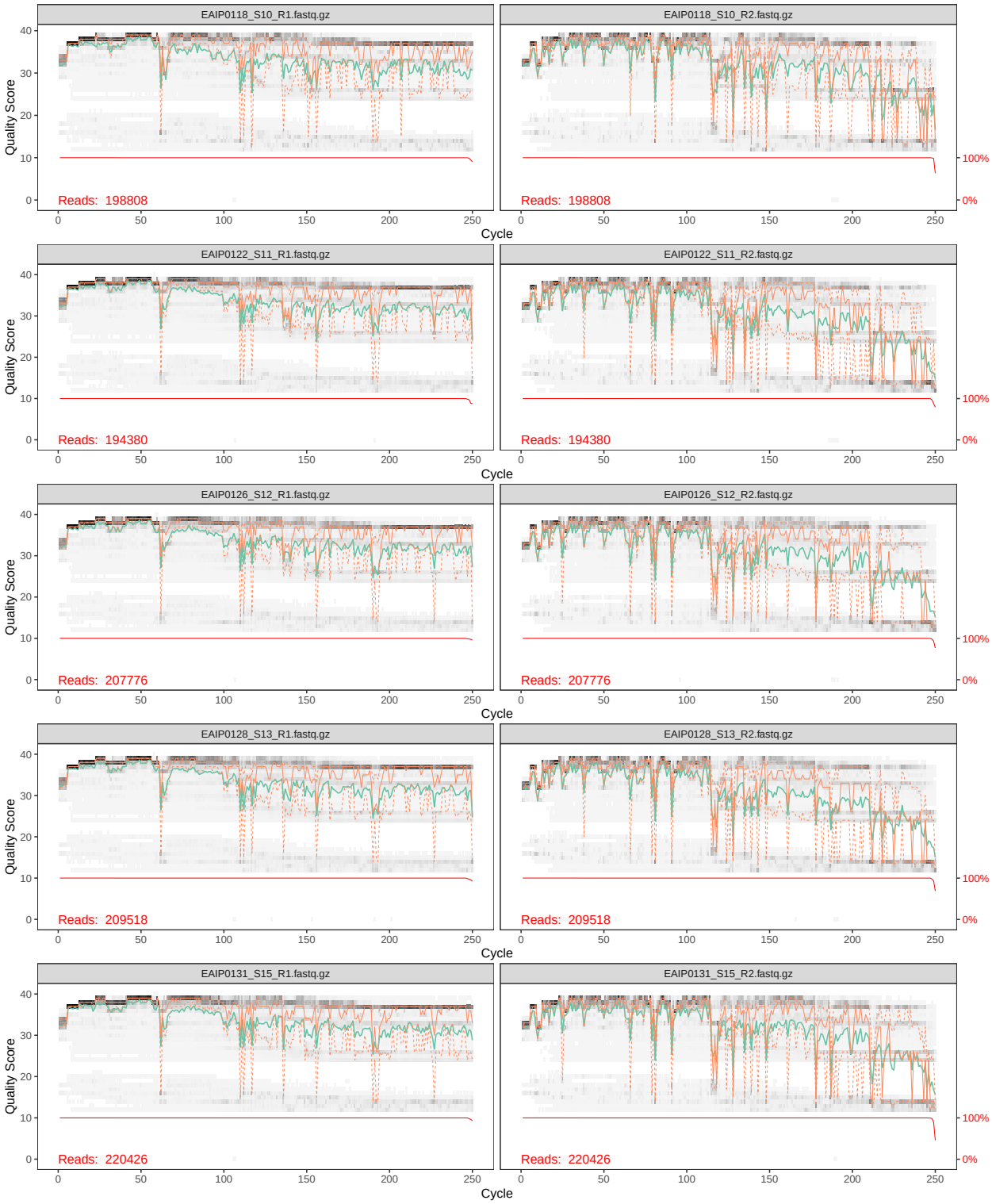

ACIP0080_S47_R1.fastq.gz
ACIP0080_S47_R2.fastq.gz
ACIP0082_S48_R1.fastq.gz
ACIP0082_S48_R2.fastq.gz
ACIP0093_S49_R1.fastq.gz
ACIP0093_S49_R2.fastq.gz
ACIP0107_S50_R1.fastq.gz
ACIP0107_S50_R2.fastq.gz
ACIP0112_S51_R1.fastq.gz
ACIP0112_S51_R2.fastq.gz
Quality Score
Cycle
Reads: 276447
Reads: 215189
Reads: 233884
Reads: 216024
Reads: 232745
100%
0%

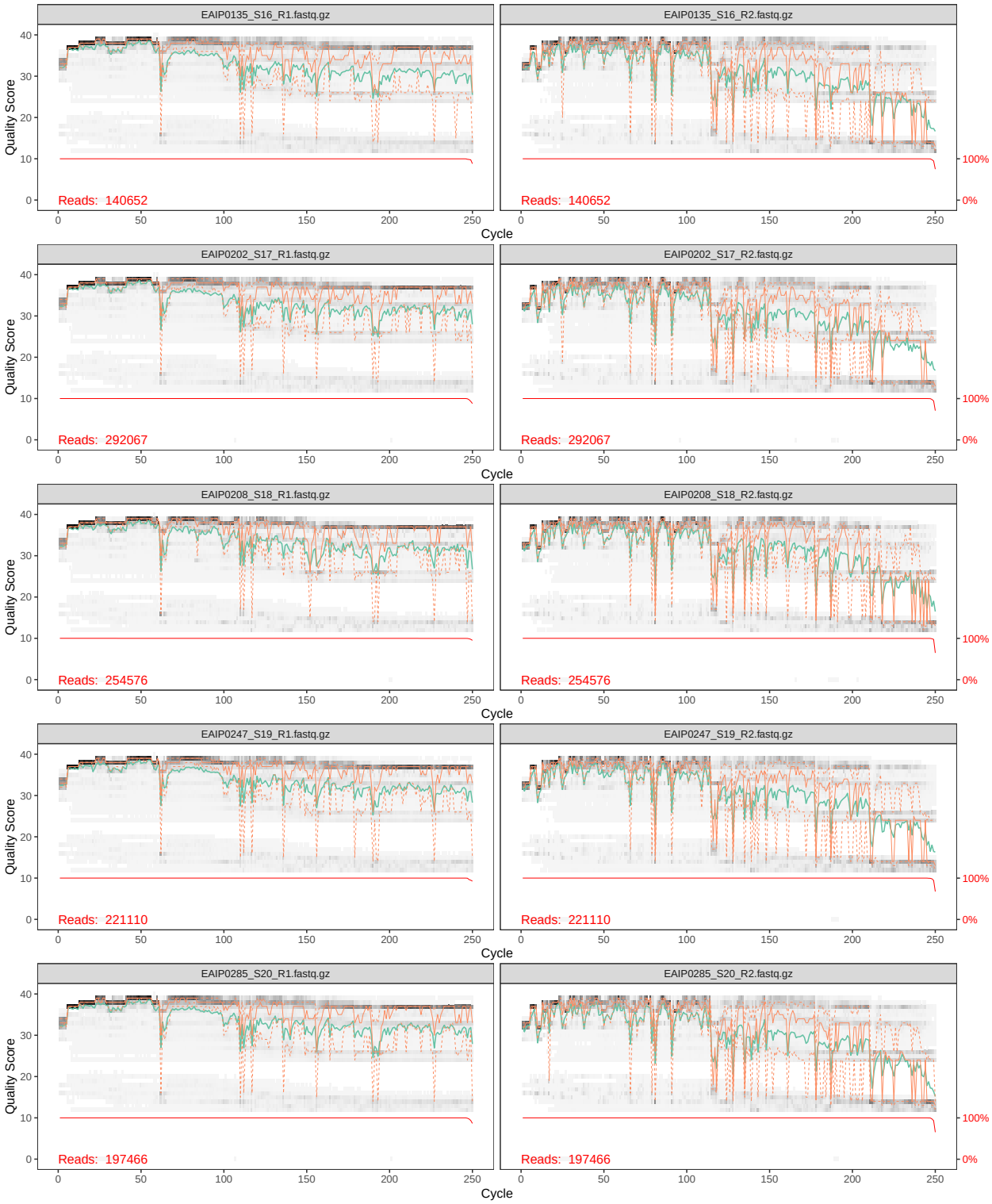

ACIP0123_S52_R1.fastq.gz
ACIP0123_S52_R2.fastq.gz
Reads: 226376
Reads: 226376
ATCCCONTROL_S55_R1.fastq.gz
ATCCCONTROL_S55_R2.fastq.gz
Reads: 228612
Reads: 228612
BAIP0040_S53_R1.fastq.gz
BAIP0040_S53_R2.fastq.gz
Reads: 191573
Reads: 191573
BLANK1_S14_R1.fastq.gz
BLANK1_S14_R2.fastq.gz
Reads: 9216
Reads: 9216
BLANK2_S28_R1.fastq.gz
BLANK2_S28_R2.fastq.gz
Reads: 11678
Reads: 11678
Quality Score
Cycle
100%
0%

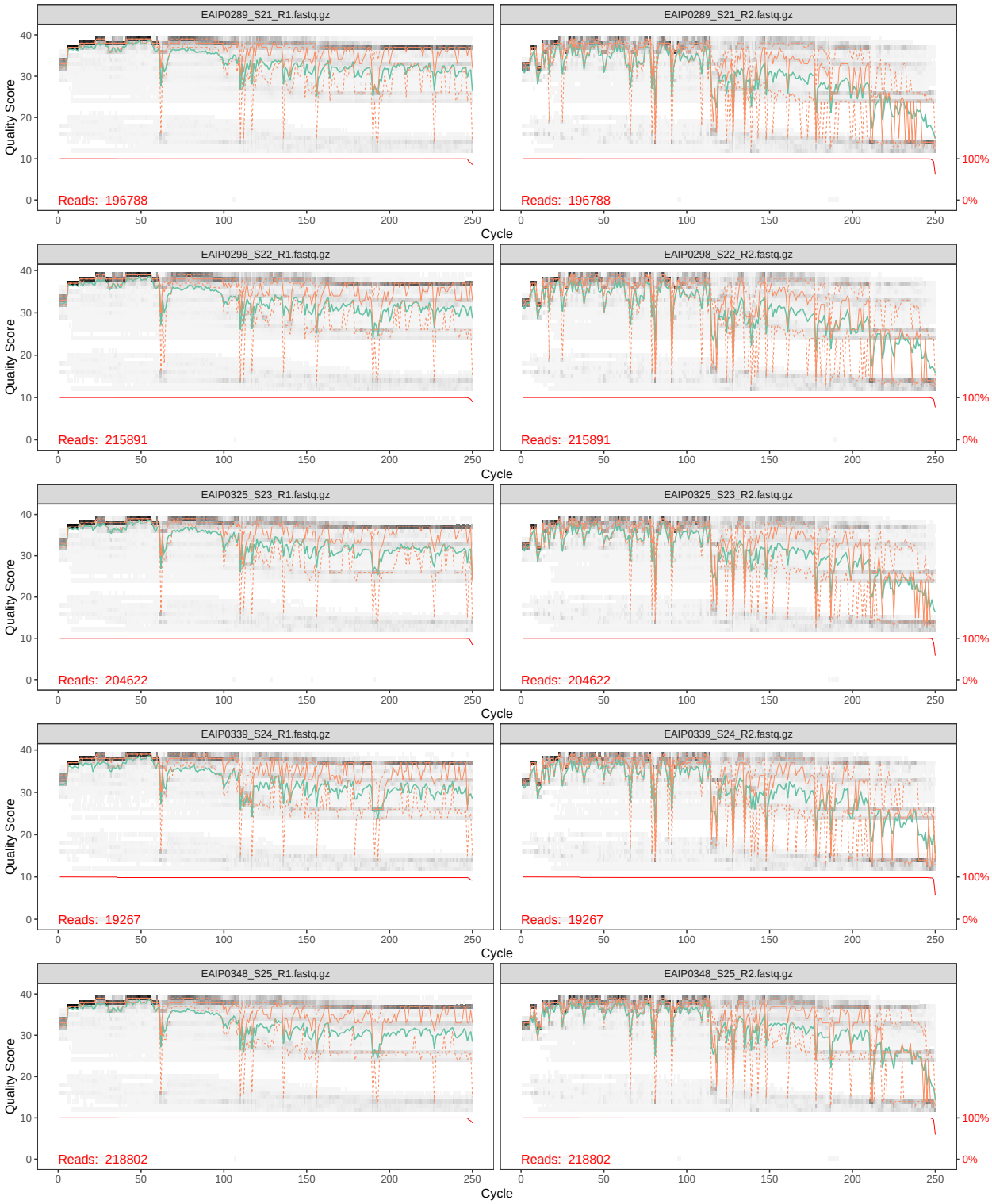

BLANK3_S42_R1.fastq.gz
BLANK3_S42_R2.fastq.gz
BLANK4_S1_R1.fastq.gz
BLANK4_S1_R2.fastq.gz
BLANK5_S15_R1.fastq.gz
BLANK5_S15_R2.fastq.gz
BLANK6_S29_R1.fastq.gz
BLANK6_S29_R2.fastq.gz
BLANK7_S43_R1.fastq.gz
BLANK7_S43_R2.fastq.gz
Quality Score
Cycle
Reads: 9286
Reads: 15573
Reads: 22601
Reads: 7363
Reads: 12662
100%
0%

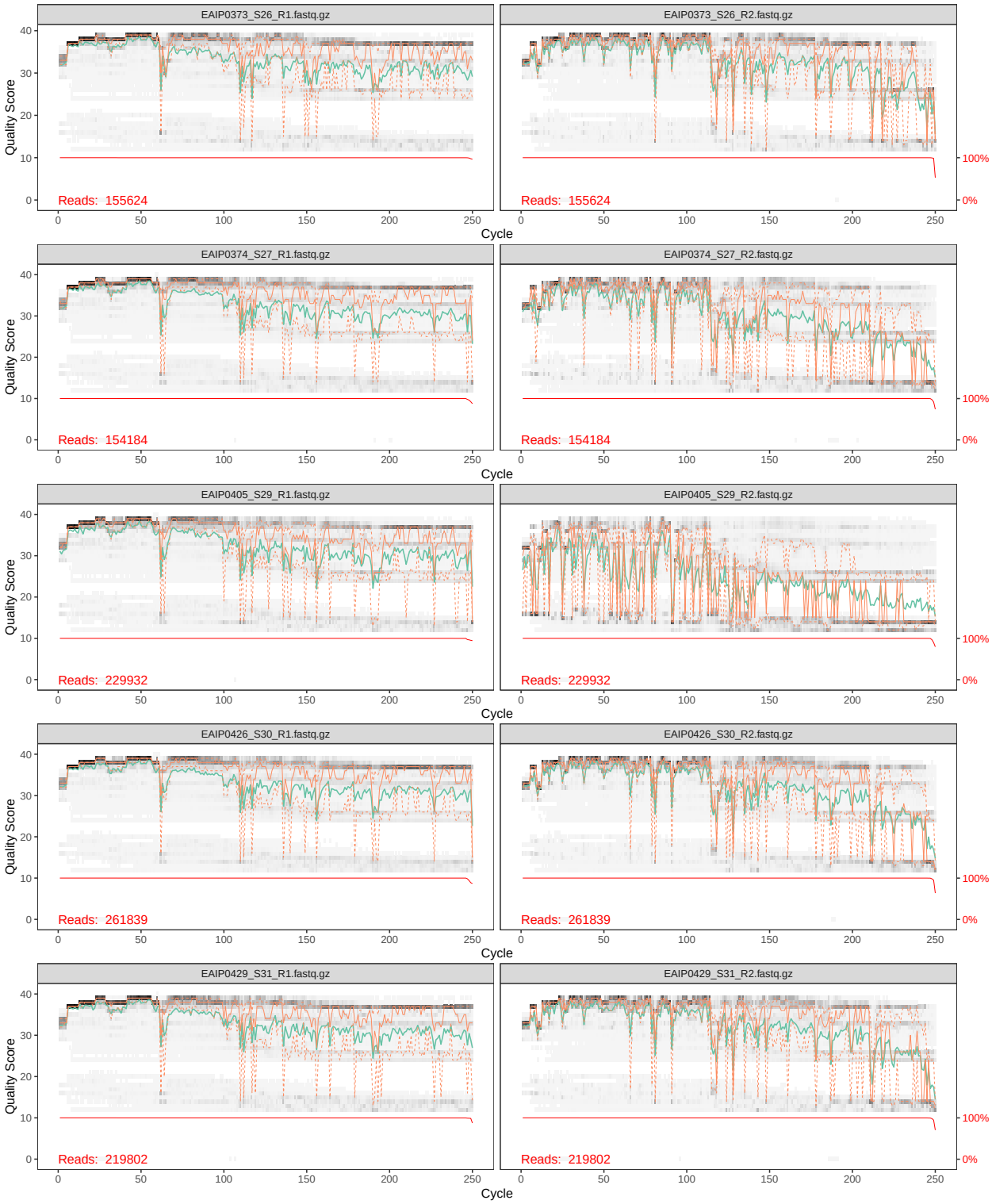

BLANK8_S54_R1.fastq.gz
BLANK8_S54_R2.fastq.gz
Reads: 7733
Reads: 7733
EAIP0027_S1_R1.fastq.gz
EAIP0027_S1_R2.fastq.gz
Reads: 214879
Reads: 214879
EAIP0028_S2_R1.fastq.gz
EAIP0028_S2_R2.fastq.gz
Reads: 241822
Reads: 241822
EAIP0029_S3_R1.fastq.gz
EAIP0029_S3_R2.fastq.gz
Reads: 255635
Reads: 255635
EAIP0032_S4_R1.fastq.gz
EAIP0032_S4_R2.fastq.gz
Reads: 14378
Reads: 14378
Quality Score
Cycle
100%
0%

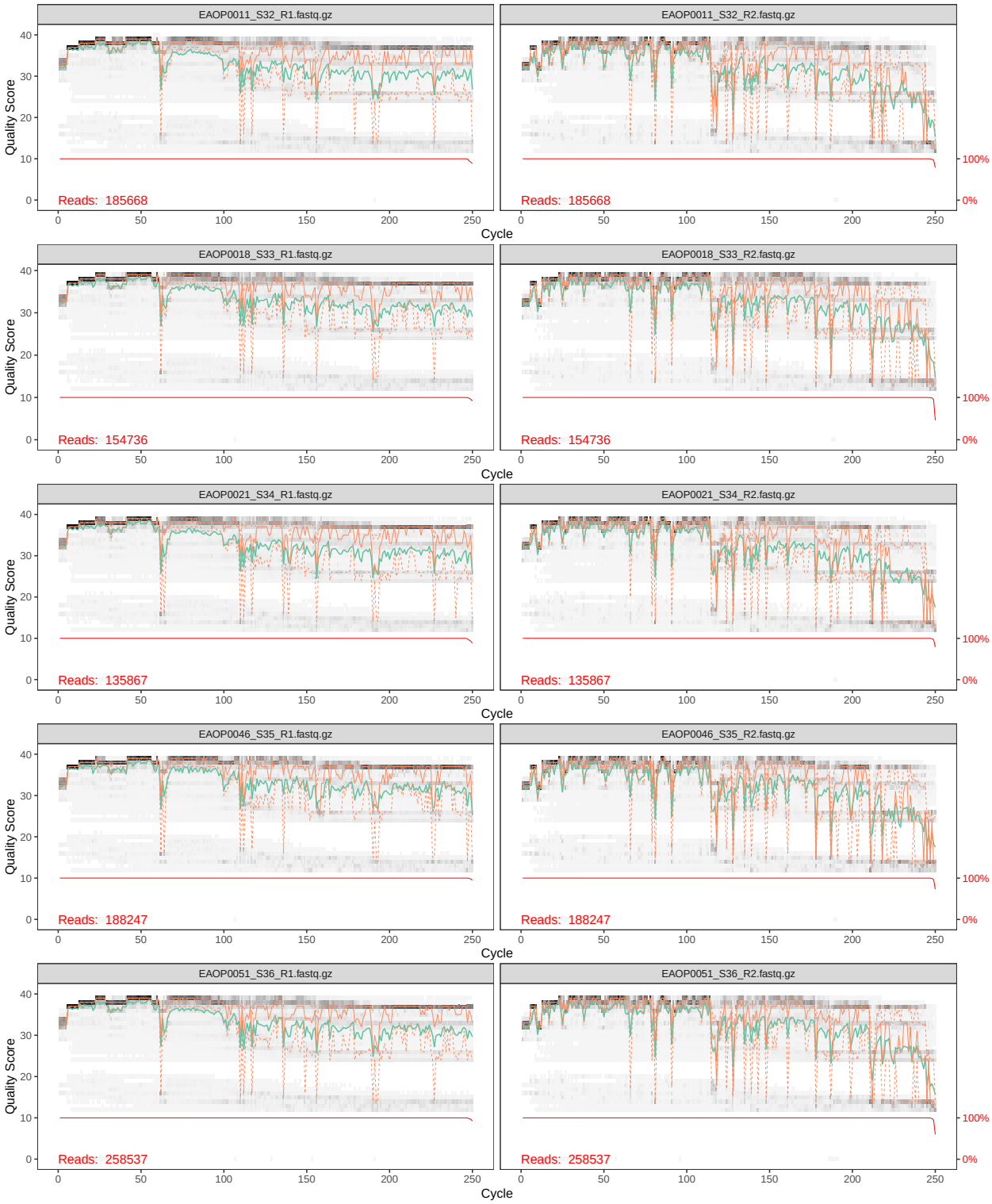

EAIP0067_S5_R1.fastq.gz
EAIP0067_S5_R2.fastq.gz
Reads: 287613
Reads: 287613
EAIP0074_S6_R1.fastq.gz
EAIP0074_S6_R2.fastq.gz
Reads: 294855
Reads: 294855
EAIP0083_S7_R1.fastq.gz
EAIP0083_S7_R2.fastq.gz
Reads: 240451
Reads: 240451
EAIP0094_S8_R1.fastq.gz
EAIP0094_S8_R2.fastq.gz
Reads: 260911
Reads: 260911
EAIP0104_S9_R1.fastq.gz
EAIP0104_S9_R2.fastq.gz
Reads: 166735
Reads: 166735
Quality Score
Cycle
100%
0%

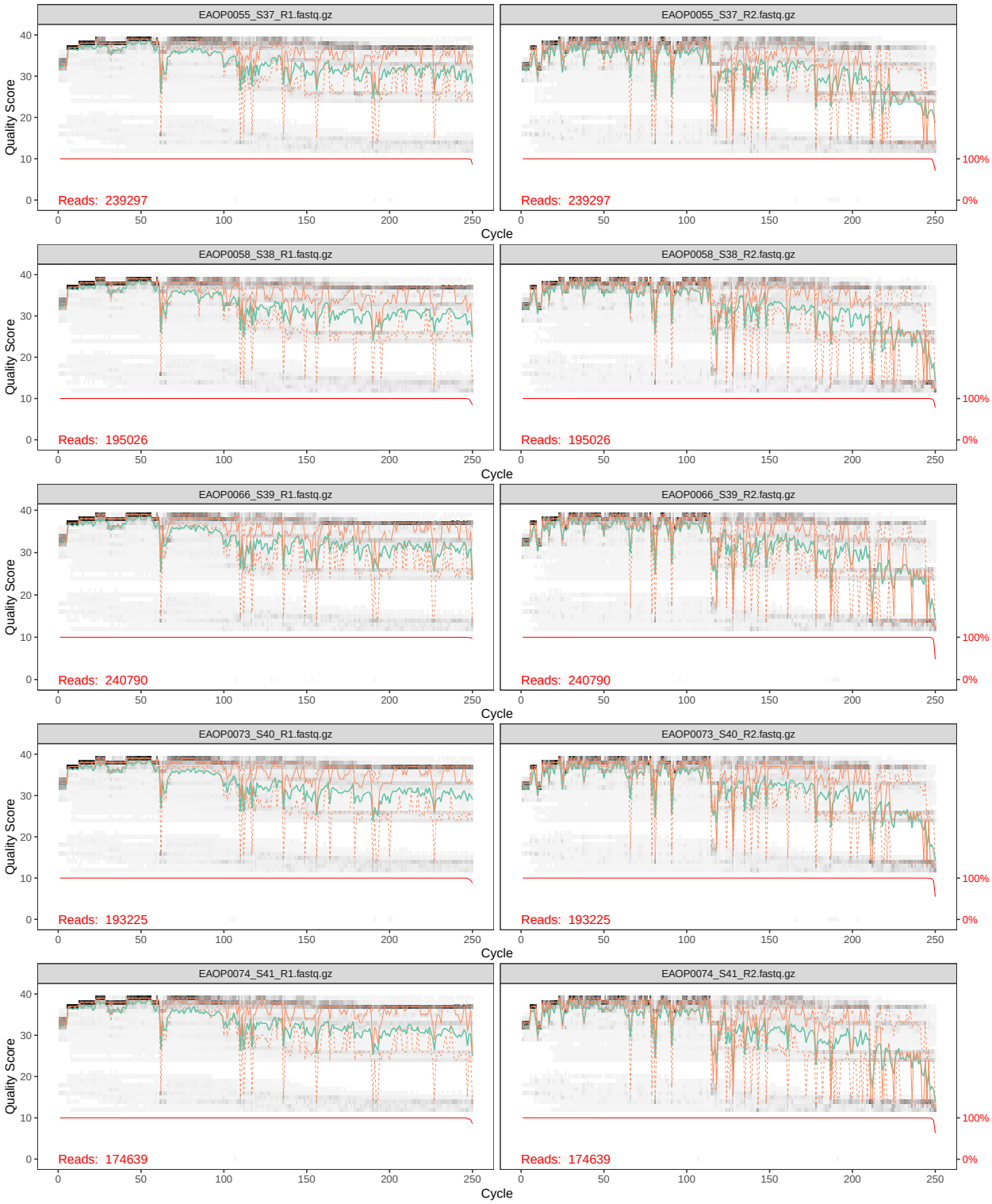

EAIP0118_S10_R1.fastq.gz
EAIP0118_S10_R2.fastq.gz
Reads: 198808
Reads: 198808
EAIP0122_S11_R1.fastq.gz
EAIP0122_S11_R2.fastq.gz
Reads: 194380
Reads: 194380
EAIP0126_S12_R1.fastq.gz
EAIP0126_S12_R2.fastq.gz
Reads: 207776
Reads: 207776
EAIP0128_S13_R1.fastq.gz
EAIP0128_S13_R2.fastq.gz
Reads: 209518
Reads: 209518
EAIP0131_S15_R1.fastq.gz
EAIP0131_S15_R2.fastq.gz
Reads: 220426
Reads: 220426
Quality Score
Cycle
100%
0%

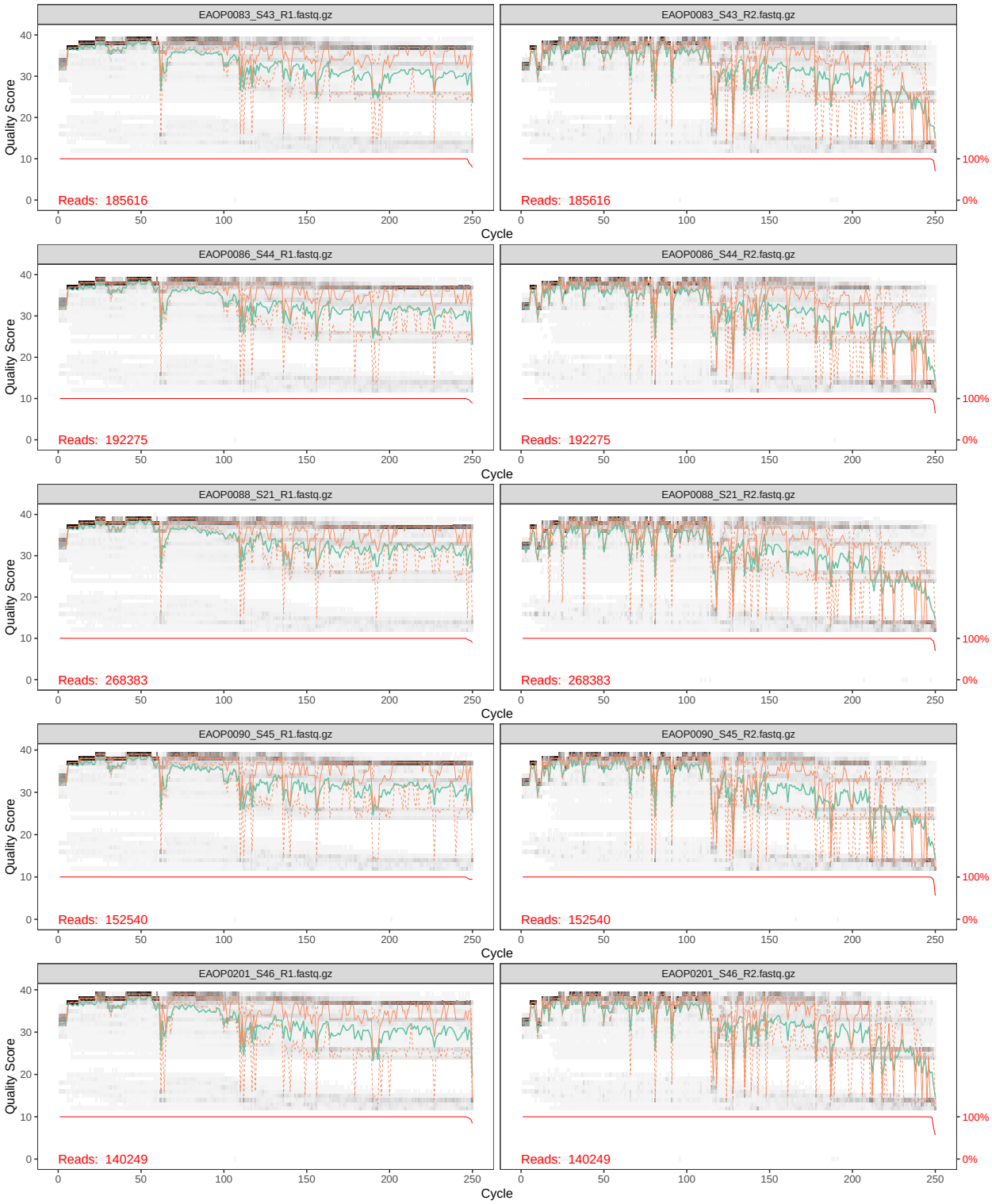

EAIP0135_S16_R1.fastq.gz
EAIP0135_S16_R2.fastq.gz
Reads: 140652
Reads: 140652
EAIP0202_S17_R1.fastq.gz
EAIP0202_S17_R2.fastq.gz
Reads: 292067
Reads: 292067
EAIP0208_S18_R1.fastq.gz
EAIP0208_S18_R2.fastq.gz
Reads: 254576
Reads: 254576
EAIP0247_S19_R1.fastq.gz
EAIP0247_S19_R2.fastq.gz
Reads: 221110
Reads: 221110
EAIP0285_S20_R1.fastq.gz
EAIP0285_S20_R2.fastq.gz
Reads: 197466
Reads: 197466
Quality Score
Cycle
100%
0%

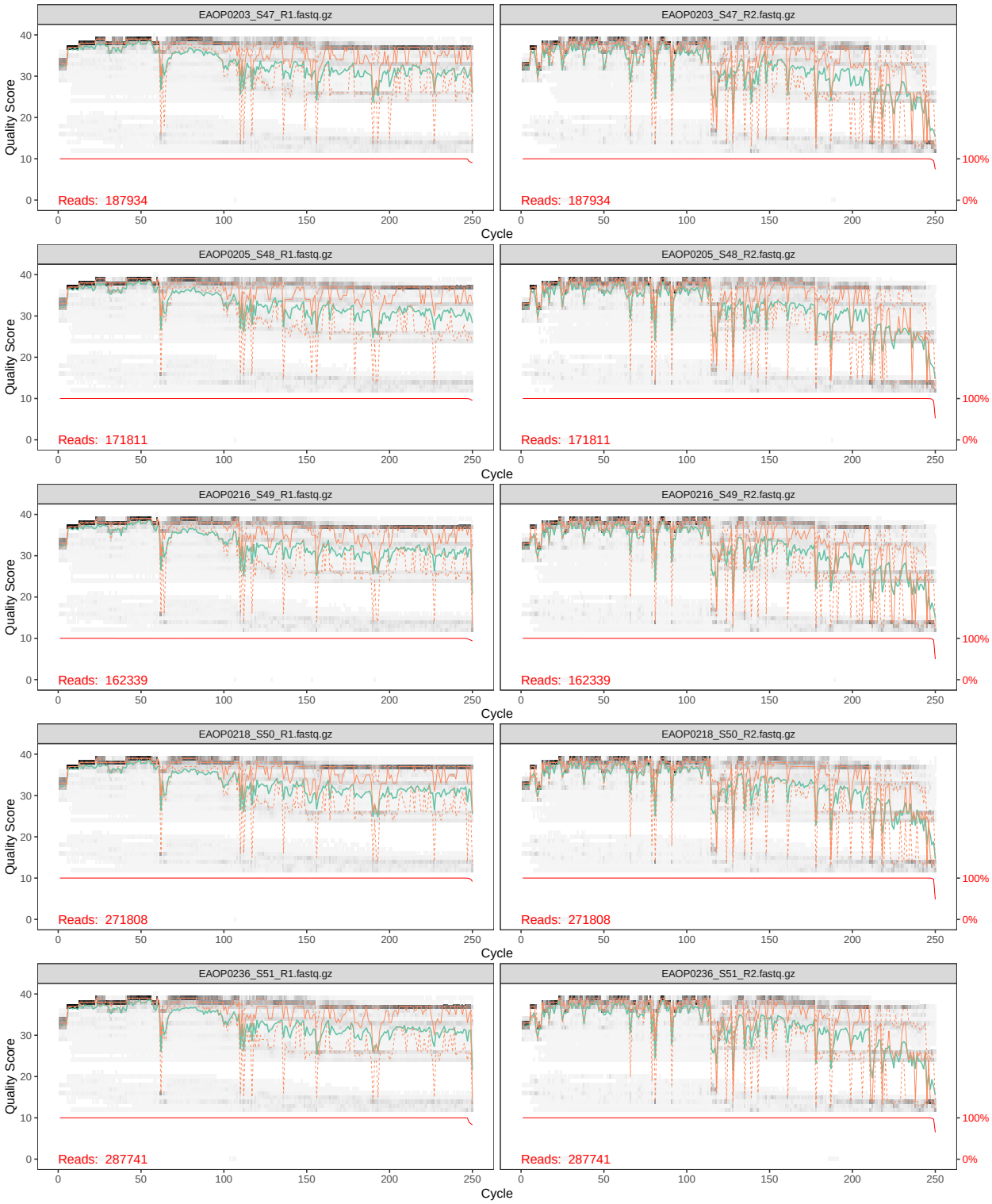

EAIP0289_S21_R1.fastq.gz
EAIP0289_S21_R2.fastq.gz
Reads: 196788
Reads: 196788
EAIP0298_S22_R1.fastq.gz
EAIP0298_S22_R2.fastq.gz
Reads: 215891
Reads: 215891
EAIP0325_S23_R1.fastq.gz
EAIP0325_S23_R2.fastq.gz
Reads: 204622
Reads: 204622
EAIP0339_S24_R1.fastq.gz
EAIP0339_S24_R2.fastq.gz
Reads: 19267
Reads: 19267
EAIP0348_S25_R1.fastq.gz
EAIP0348_S25_R2.fastq.gz
Reads: 218802
Reads: 218802
Quality Score
Cycle
100%
0%

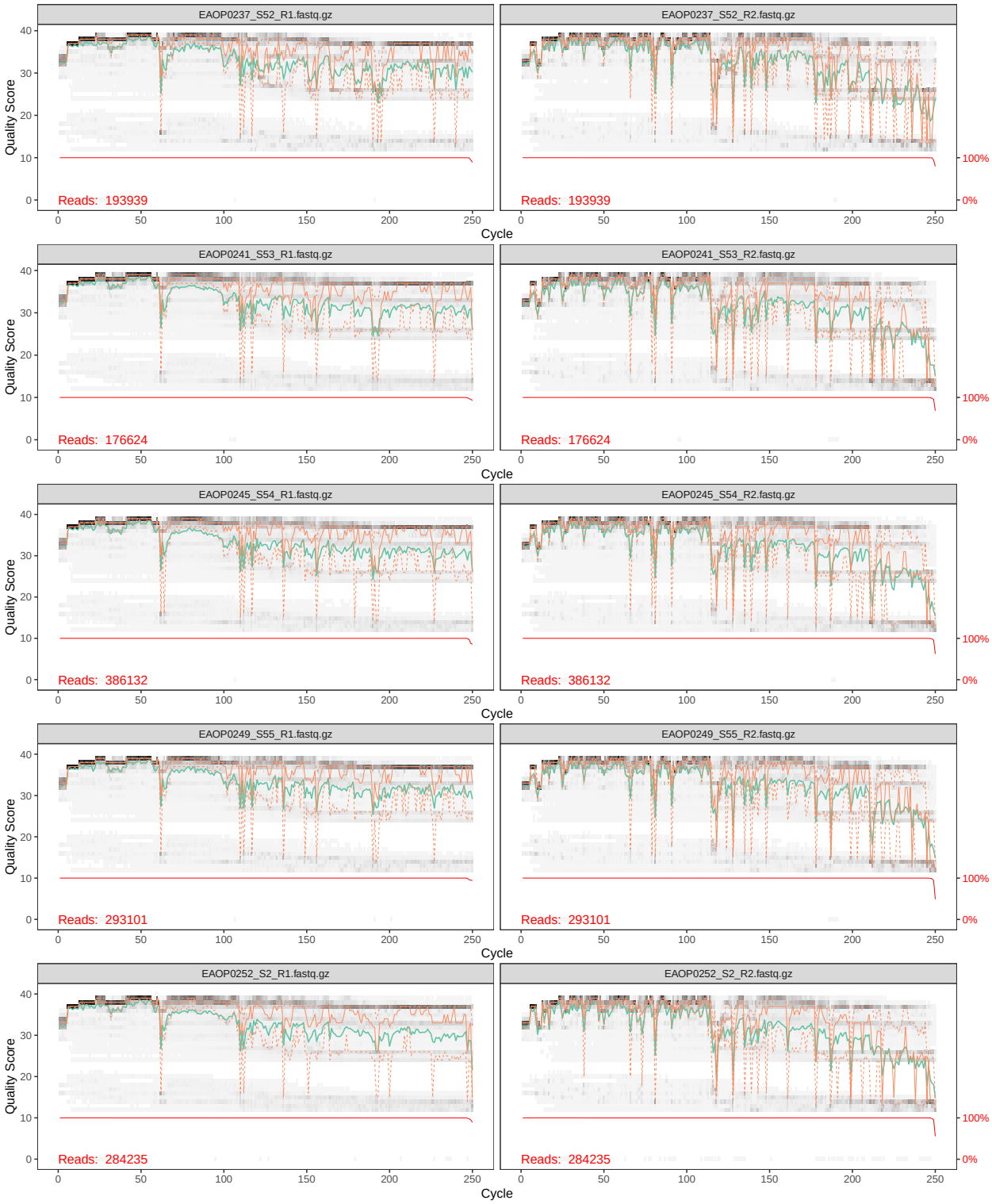

EAIP0373_S26_R1.fastq.gz
EAIP0373_S26_R2.fastq.gz
Reads: 155624
Reads: 155624
EAIP0374_S27_R1.fastq.gz
EAIP0374_S27_R2.fastq.gz
Reads: 154184
Reads: 154184
EAIP0405_S29_R1.fastq.gz
EAIP0405_S29_R2.fastq.gz
Reads: 229932
Reads: 229932
EAIP0426_S30_R1.fastq.gz
EAIP0426_S30_R2.fastq.gz
Reads: 261839
Reads: 261839
EAIP0429_S31_R1.fastq.gz
EAIP0429_S31_R2.fastq.gz
Reads: 219802
Reads: 219802
Quality Score
Cycle
100%
0%

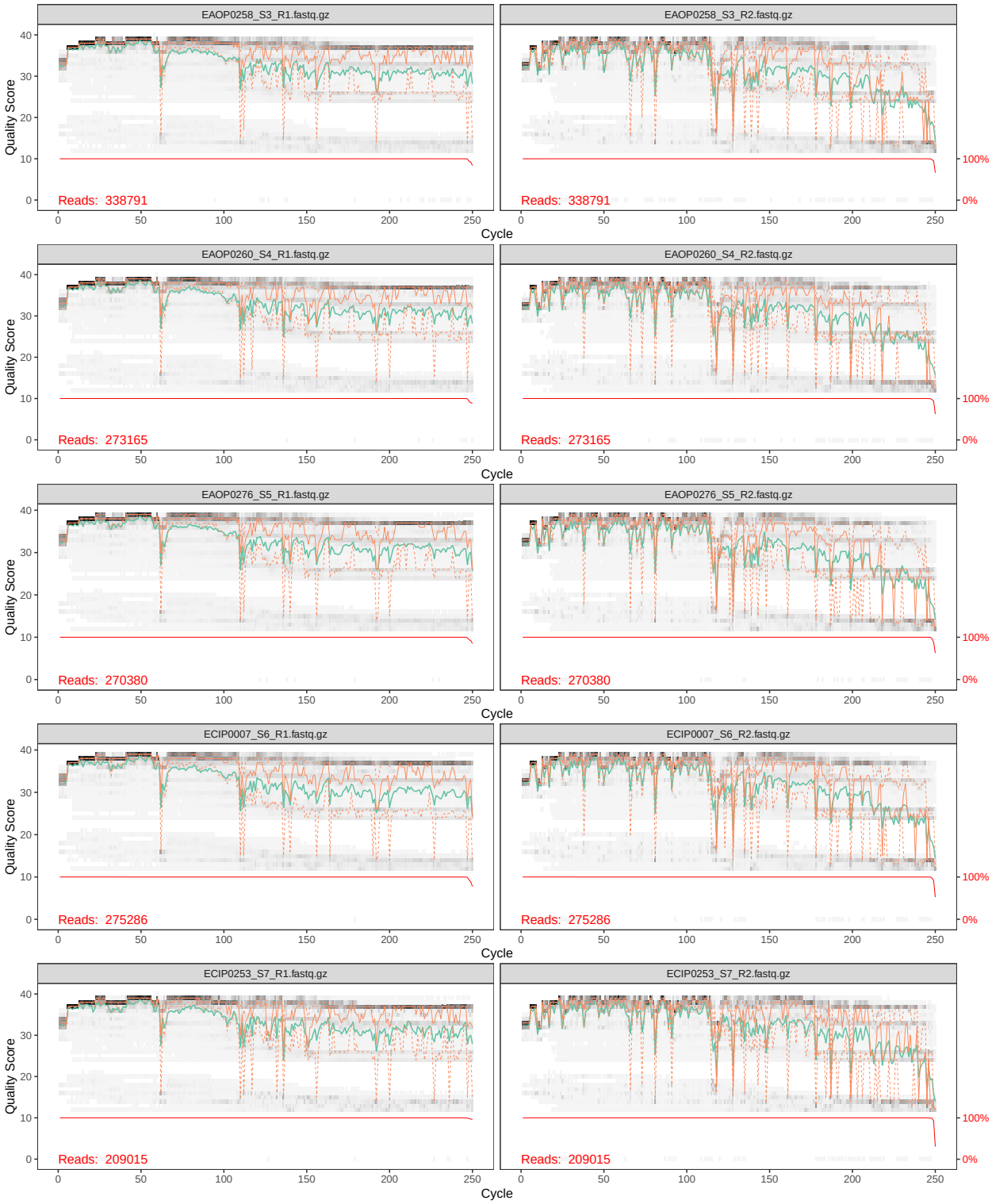

EAOP0011_S32_R1.fastq.gz
EAOP0011_S32_R2.fastq.gz
Reads: 185668
Reads: 185668
EAOP0018_S33_R1.fastq.gz
EAOP0018_S33_R2.fastq.gz
Reads: 154736
Reads: 154736
EAOP0021_S34_R1.fastq.gz
EAOP0021_S34_R2.fastq.gz
Reads: 135867
Reads: 135867
EAOP0046_S35_R1.fastq.gz
EAOP0046_S35_R2.fastq.gz
Reads: 188247
Reads: 188247
EAOP0051_S36_R1.fastq.gz
EAOP0051_S36_R2.fastq.gz
Reads: 258537
Reads: 258537
Quality Score
Cycle
100%
0%

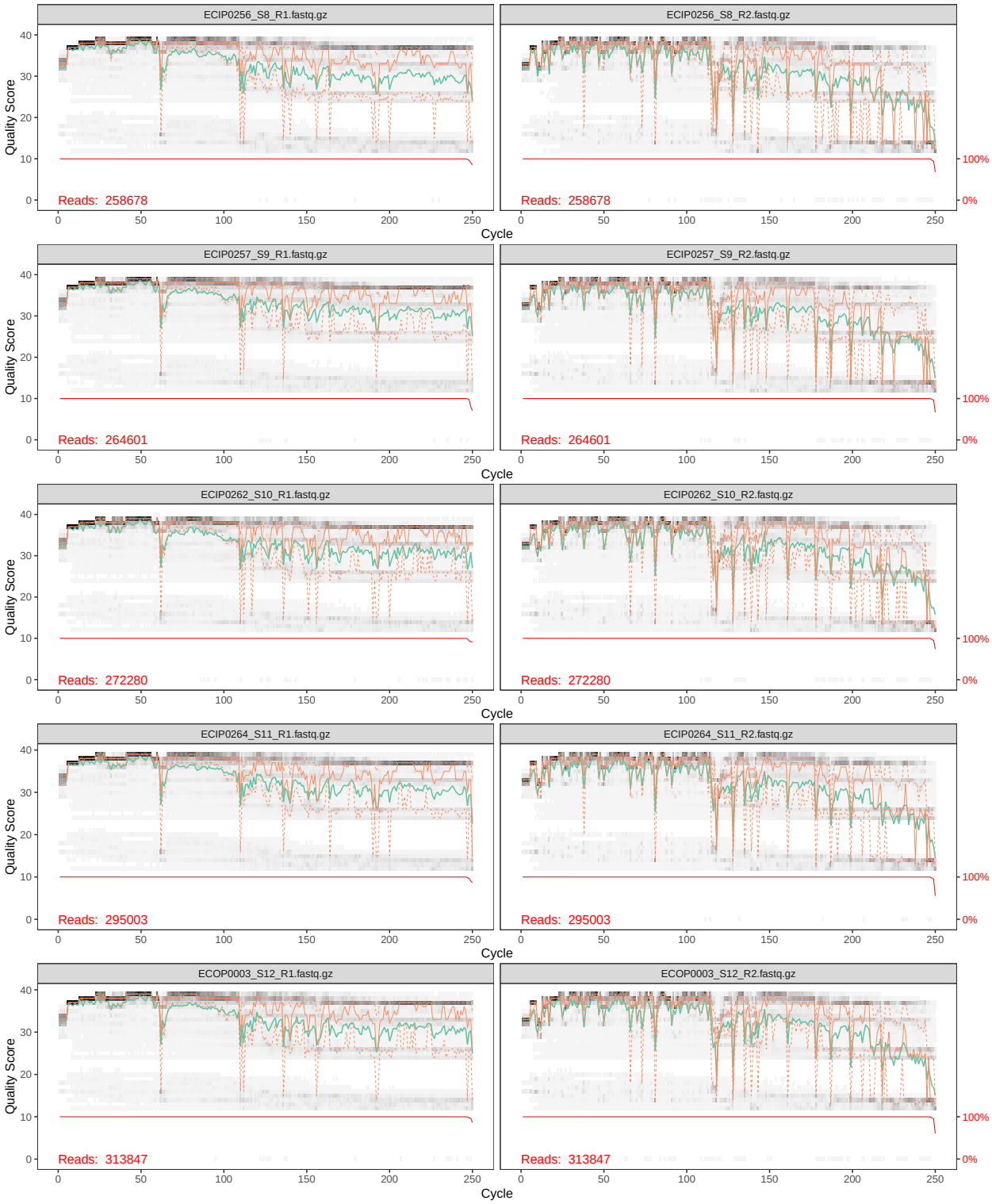

EAOP0055_S37_R1.fastq.gz
EAOP0055_S37_R2.fastq.gz
Reads: 239297
Reads: 239297
EAOP0058_S38_R1.fastq.gz
EAOP0058_S38_R2.fastq.gz
Reads: 195026
Reads: 195026
EAOP0066_S39_R1.fastq.gz
EAOP0066_S39_R2.fastq.gz
Reads: 240790
Reads: 240790
EAOP0073_S40_R1.fastq.gz
EAOP0073_S40_R2.fastq.gz
Reads: 193225
Reads: 193225
EAOP0074_S41_R1.fastq.gz
EAOP0074_S41_R2.fastq.gz
Reads: 174639
Reads: 174639
Quality Score
Cycle
100%
0%

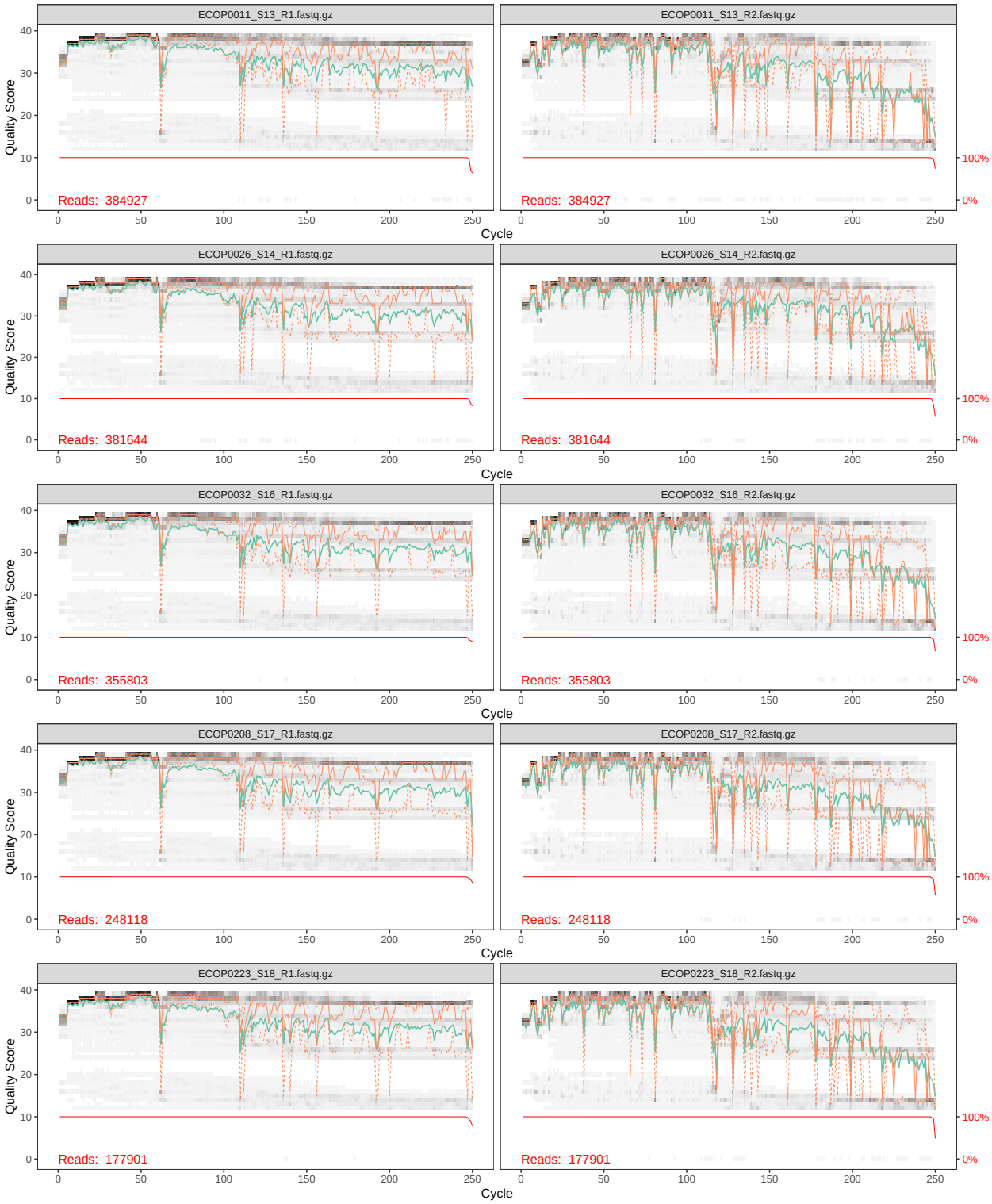

EAOP0083_S43_R1.fastq.gz
EAOP0083_S43_R2.fastq.gz
EAOP0086_S44_R1.fastq.gz
EAOP0086_S44_R2.fastq.gz
EAOP0088_S21_R1.fastq.gz
EAOP0088_S21_R2.fastq.gz
EAOP0090_S45_R1.fastq.gz
EAOP0090_S45_R2.fastq.gz
EAOP0201_S46_R1.fastq.gz
EAOP0201_S46_R2.fastq.gz
Quality Score
Cycle
Reads: 185616
Reads: 192275
Reads: 268383
Reads: 152540
Reads: 140249
100%
0%

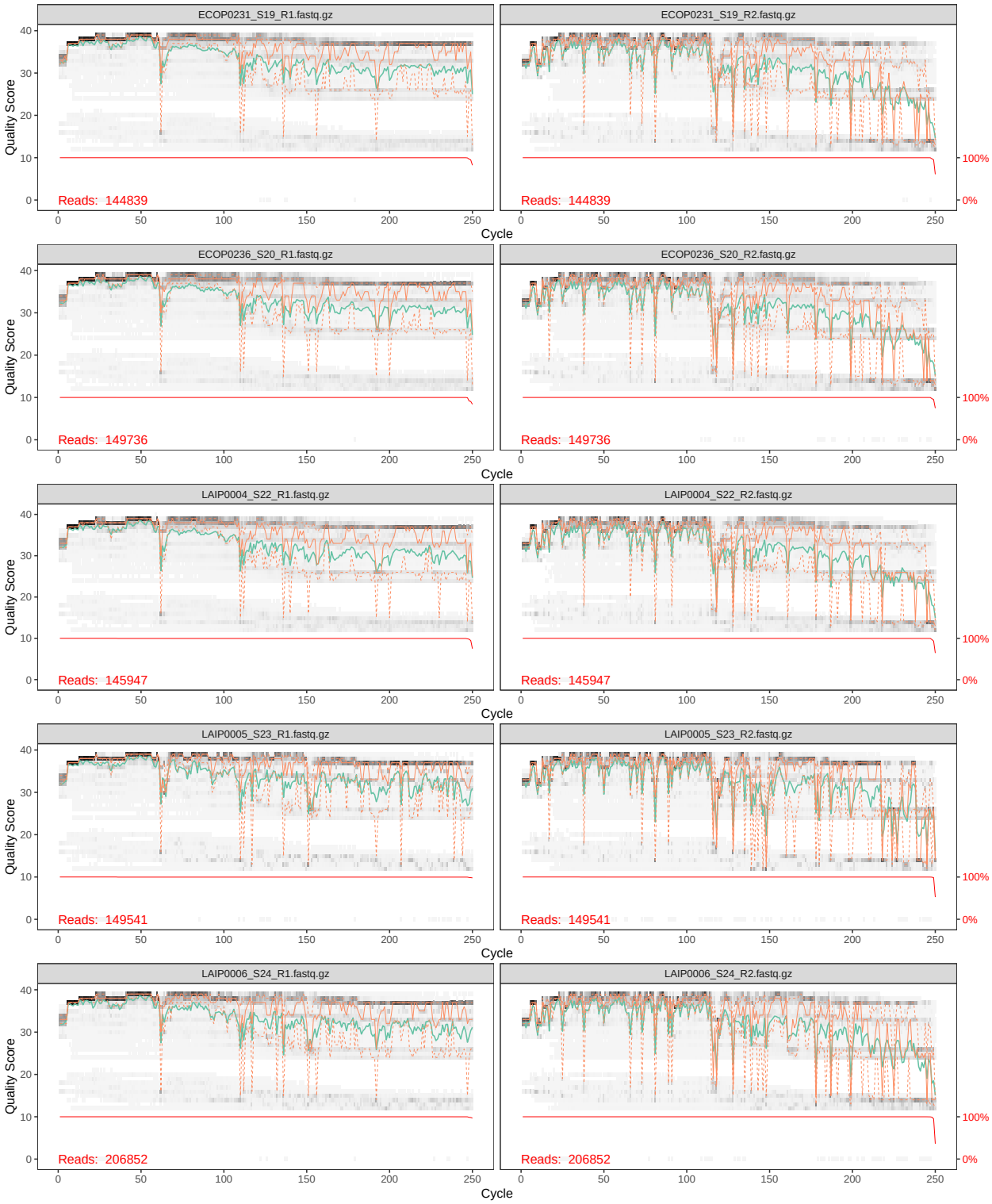

EAOP0203_S47_R1.fastq.gz
EAOP0203_S47_R2.fastq.gz
Reads: 187934
Reads: 187934
EAOP0205_S48_R1.fastq.gz
EAOP0205_S48_R2.fastq.gz
Reads: 171811
Reads: 171811
EAOP0216_S49_R1.fastq.gz
EAOP0216_S49_R2.fastq.gz
Reads: 162339
Reads: 162339
EAOP0218_S50_R1.fastq.gz
EAOP0218_S50_R2.fastq.gz
Reads: 271808
Reads: 271808
EAOP0236_S51_R1.fastq.gz
EAOP0236_S51_R2.fastq.gz
Reads: 287741
Reads: 287741
Quality Score
Cycle
100%
0%

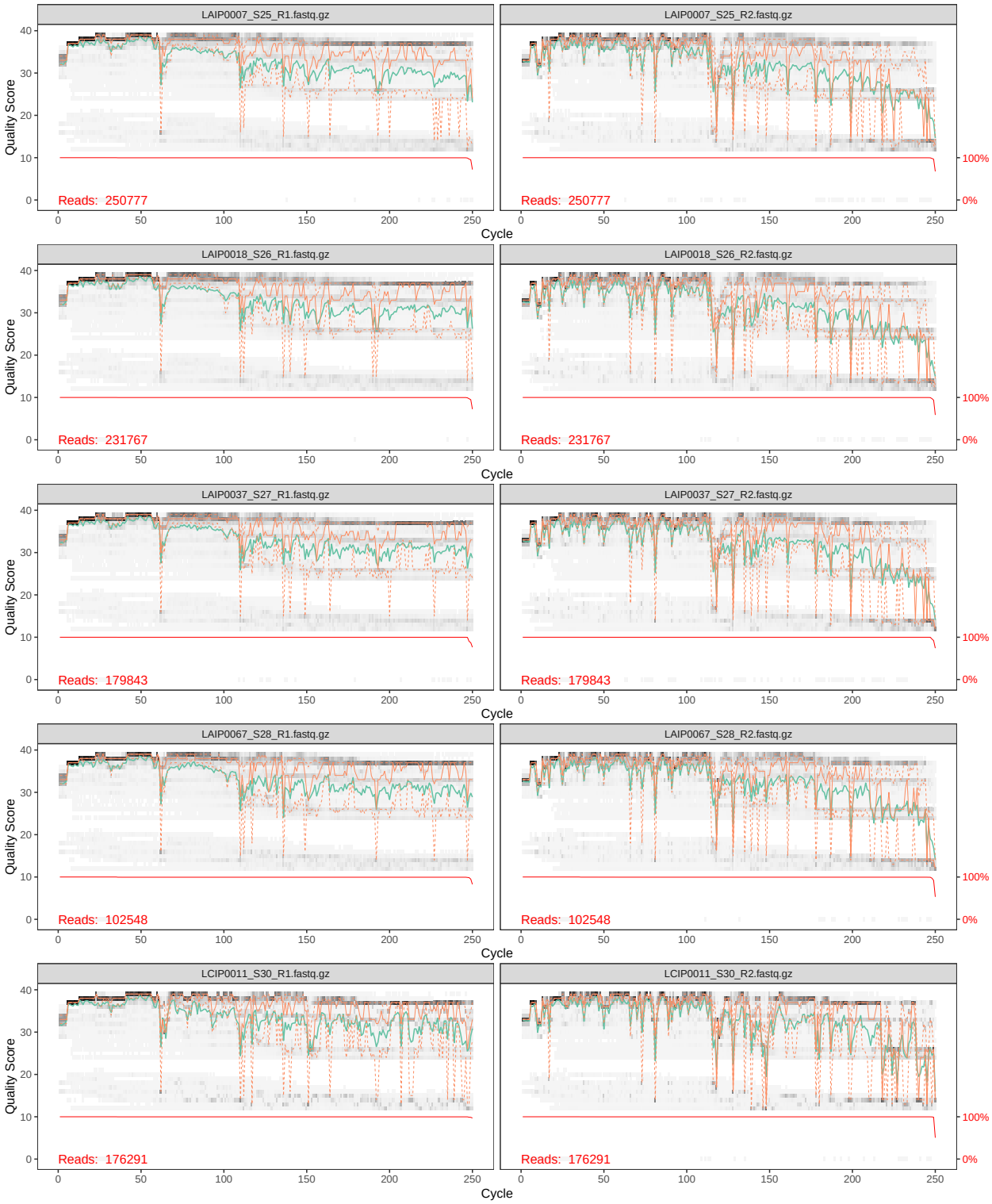

EAOP0237_S52_R1.fastq.gz
EAOP0237_S52_R2.fastq.gz
Reads: 193939
Reads: 193939
EAOP0241_S53_R1.fastq.gz
EAOP0241_S53_R2.fastq.gz
Reads: 176624
Reads: 176624
EAOP0245_S54_R1.fastq.gz
EAOP0245_S54_R2.fastq.gz
Reads: 386132
Reads: 386132
EAOP0249_S55_R1.fastq.gz
EAOP0249_S55_R2.fastq.gz
Reads: 293101
Reads: 293101
EAOP0252_S2_R1.fastq.gz
EAOP0252_S2_R2.fastq.gz
Reads: 284235
Reads: 284235
Quality Score
Cycle
100%
0%

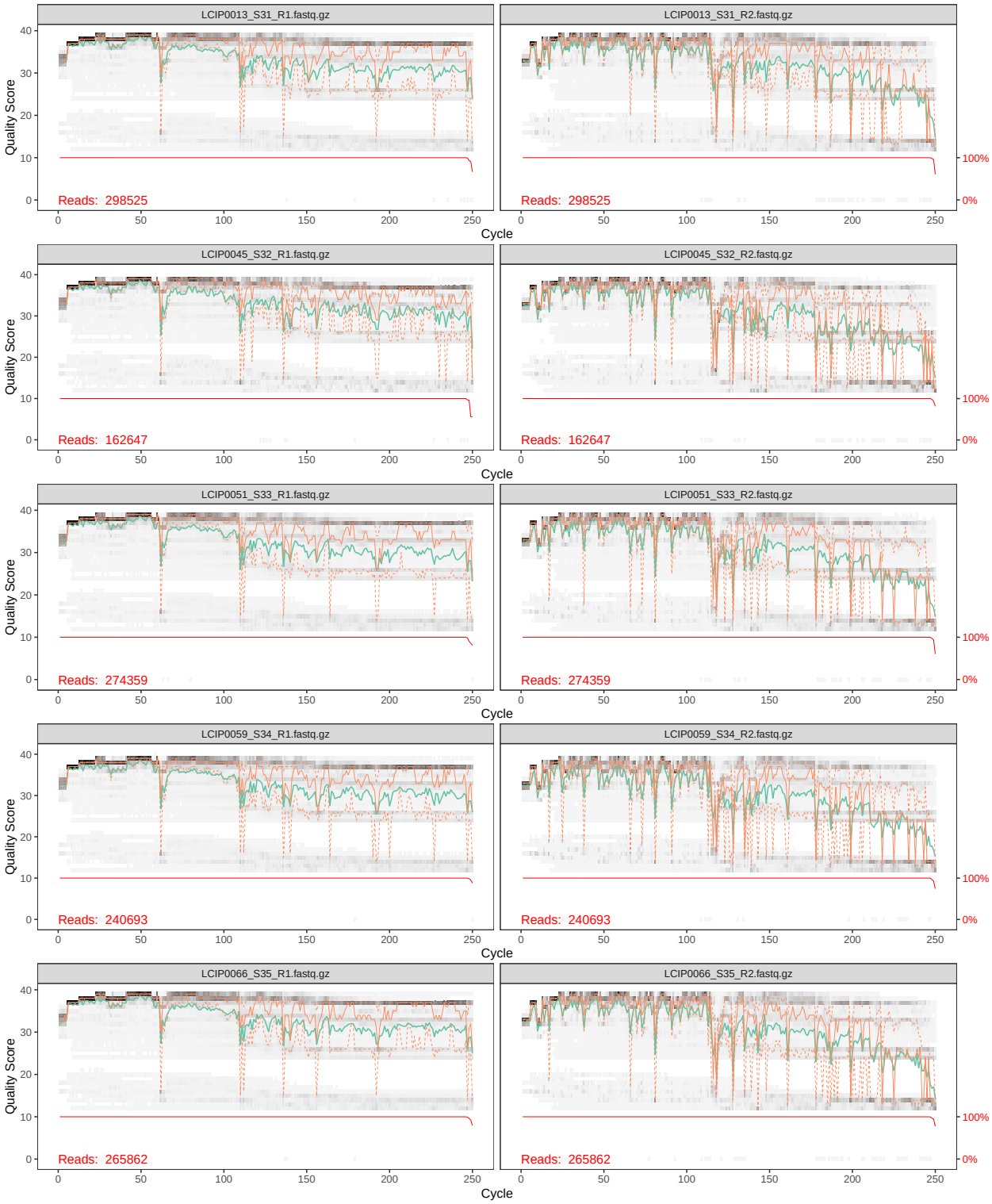

EAOP0258_S3_R1.fastq.gz
EAOP0258_S3_R2.fastq.gz
Reads: 338791
Reads: 338791
EAOP0260_S4_R1.fastq.gz
EAOP0260_S4_R2.fastq.gz
Reads: 273165
Reads: 273165
EAOP0276_S5_R1.fastq.gz
EAOP0276_S5_R2.fastq.gz
Reads: 270380
Reads: 270380
ECIP0007_S6_R1.fastq.gz
ECIP0007_S6_R2.fastq.gz
Reads: 275286
Reads: 275286
ECIP0253_S7_R1.fastq.gz
ECIP0253_S7_R2.fastq.gz
Reads: 209015
Reads: 209015
Quality Score
Cycle
100%
0%

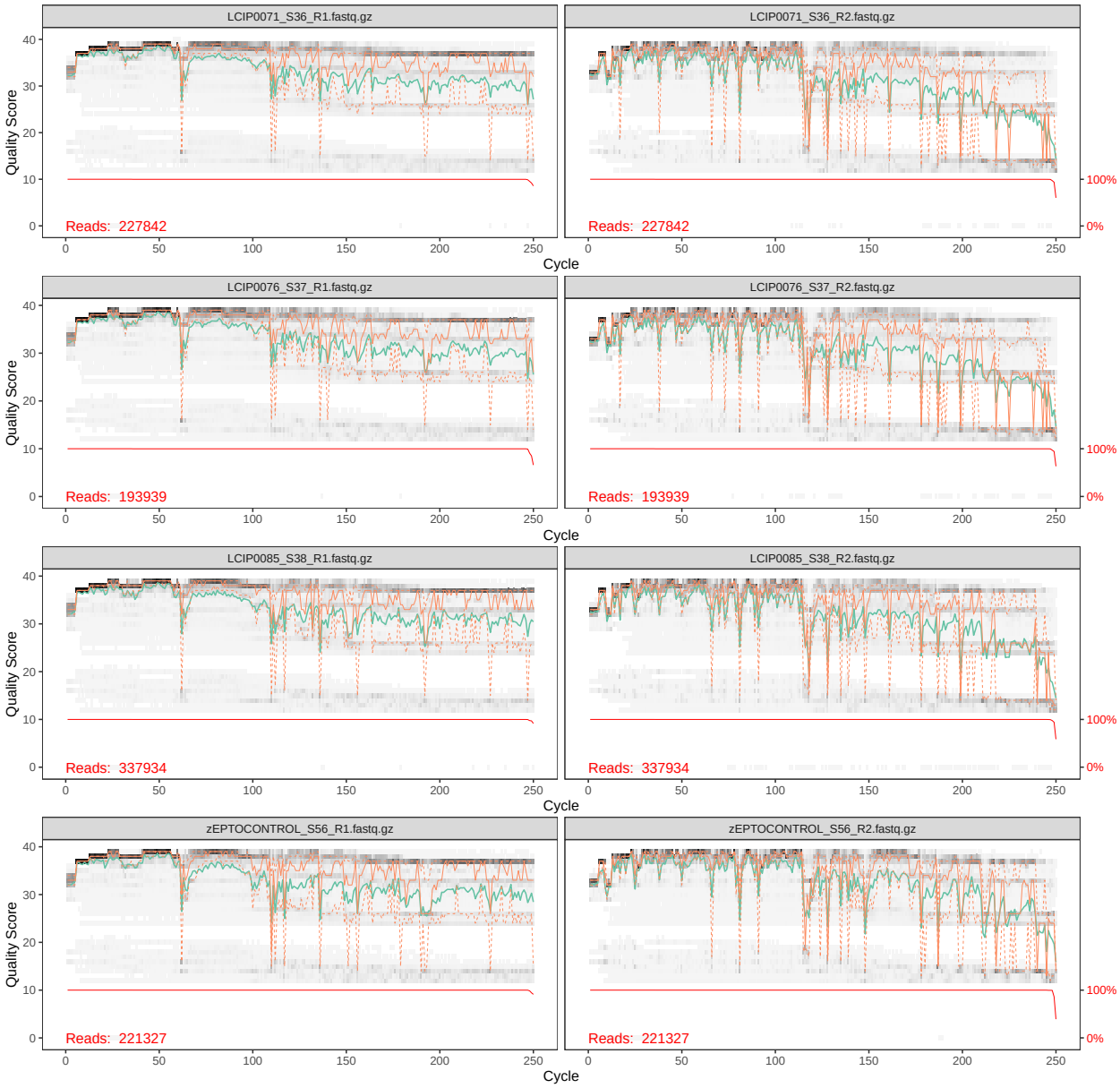

ECIP0256_S8_R1.fastq.gz
ECIP0256_S8_R2.fastq.gz
Reads: 258678
Reads: 258678
ECIP0257_S9_R1.fastq.gz
ECIP0257_S9_R2.fastq.gz
Reads: 264601
Reads: 264601
ECIP0262_S10_R1.fastq.gz
ECIP0262_S10_R2.fastq.gz
Reads: 272280
Reads: 272280
ECIP0264_S11_R1.fastq.gz
ECIP0264_S11_R2.fastq.gz
Reads: 295003
Reads: 295003
ECOP0003_S12_R1.fastq.gz
ECOP0003_S12_R2.fastq.gz
Reads: 313847
Reads: 313847
Quality Score
Cycle
100%
0%


ECOP0011_S13_R1.fastq.gz
ECOP0011_S13_R2.fastq.gz
Reads: 384927
Reads: 384927
ECOP0026_S14_R1.fastq.gz
ECOP0026_S14_R2.fastq.gz
Reads: 381644
Reads: 381644
ECOP0032_S16_R1.fastq.gz
ECOP0032_S16_R2.fastq.gz
Reads: 355803
Reads: 355803
ECOP0208_S17_R1.fastq.gz
ECOP0208_S17_R2.fastq.gz
Reads: 248118
Reads: 248118
ECOP0223_S18_R1.fastq.gz
ECOP0223_S18_R2.fastq.gz
Reads: 177901
Reads: 177901
Quality Score
Cycle
100%
0%


ECOP0231_S19_R1.fastq.gz
ECOP0231_S19_R2.fastq.gz
ECOP0236_S20_R1.fastq.gz
ECOP0236_S20_R2.fastq.gz
LAIP0004_S22_R1.fastq.gz
LAIP0004_S22_R2.fastq.gz
LAIP0005_S23_R1.fastq.gz
LAIP0005_S23_R2.fastq.gz
LAIP0006_S24_R1.fastq.gz
LAIP0006_S24_R2.fastq.gz
Quality Score
Cycle
Reads: 144839
Reads: 149736
Reads: 145947
Reads: 149541
Reads: 206852
100%
0%


LAIP0007_S25_R1.fastq.gz
LAIP0007_S25_R2.fastq.gz
LAIP0018_S26_R1.fastq.gz
LAIP0018_S26_R2.fastq.gz
LAIP0037_S27_R1.fastq.gz
LAIP0037_S27_R2.fastq.gz
LAIP0067_S28_R1.fastq.gz
LAIP0067_S28_R2.fastq.gz
LCIP0011_S30_R1.fastq.gz
LCIP0011_S30_R2.fastq.gz
Quality Score
Cycle
Reads: 250777
Reads: 231767
Reads: 179843
Reads: 102548
Reads: 176291
100%
0%


LCIP0013_S31_R1.fastq.gz
LCIP0013_S31_R2.fastq.gz
Reads: 298525
Reads: 298525
LCIP0045_S32_R1.fastq.gz
LCIP0045_S32_R2.fastq.gz
Reads: 162647
Reads: 162647
LCIP0051_S33_R1.fastq.gz
LCIP0051_S33_R2.fastq.gz
Reads: 274359
Reads: 274359
LCIP0059_S34_R1.fastq.gz
LCIP0059_S34_R2.fastq.gz
Reads: 240693
Reads: 240693
LCIP0066_S35_R1.fastq.gz
LCIP0066_S35_R2.fastq.gz
Reads: 265862
Reads: 265862
Quality Score
Cycle
100%
0%


LCIP0071_S36_R1.fastq.gz
LCIP0071_S36_R2.fastq.gz
LCIP0076_S37_R1.fastq.gz
LCIP0076_S37_R2.fastq.gz
LCIP0085_S38_R1.fastq.gz
LCIP0085_S38_R2.fastq.gz
zEPTOCONTROL_S56_R1.fastq.gz
zEPTOCONTROL_S56_R2.fastq.gz
Quality Score
Cycle
Reads: 227842
Reads: 193939
Reads: 337934
Reads: 221327
100%
0%