Krause/Niazi et al. RNA Data Analysis

Adnan M. Niazi & Maximillian Krause

5/13/2019

Load the required libraries:

```
pacman::p_load(dplyr, magrittr, ggplot2, drake, knitr, ggpubr, here, tidyverse)
```

Load the data:

```
loadd(rna_kr_data)
```

About the data

`rna_kr_data` is made automatically when `drake::r_make()` command is run. It is a dataframe containing RNA data from three replicates of Oxford Nanopore Direct RNA sequencing experiments. The first two replicates were obtained using RNA-SQK001 kit with reverse transcription, while the third replicate was obtained using RNA-SQK002 sequencing kit and omitting the reverse transcription step during library preparation.

For each read, `rna_kr_data` dataframe contains:

1. Output of tailfindr
2. Output of Nanopolish
3. Barcode label of the read, and
4. Location of the poly(A) end adjacent to eGFP transcript found by alignment of the expected eGFP sequence.

The raw CSV files from which `rna_kr_data` was made are present in the `extdata/` directory. The code used to consolidate all these disparate pieces of information into a single data frame (`rna_kr_data`) is present in the `Analyses/` and `R/` directories.

Here is a description of columns:

```
knitr::kable(col_names_df)
```

Columns	Description
read_id	Unique read ID generated by MinKNOW
barcode	Barcode assigning the expected poly(A) tail length
replicate	Replicate number
file_path	Full file path (not relevant)
tail_start_tf	tailfindr's estimate of poly(A) start site
tail_end_tf	tailfindr's estimate of poly(A) end site
samples_per_nt_tf	tailfindr's estimate of read-specific nucleotide translocation rate (samples per nucleotide)
tail_length_tf	tailfindr's estimate of poly(A) tail length
tail_start_np	Nanopolish estimate of poly(A) start site
tail_end_np	Nanopolish estimate of poly(A) end site
read_rate_np	Nanopolish read rate
tail_length_np	Nanopolish estimate of poly(A) tail length
qc_tag_np	Nanopolish QC tag
samples_per_nt_np	Nanopolish estimate of read-specific nucleotide translocation rate (samples per nucleotide) ca

Columns	Description
transcript_alignment_start	Location of poly(A) end as detected by eGFP sequence alignment
kit	ONT Library prep kit used

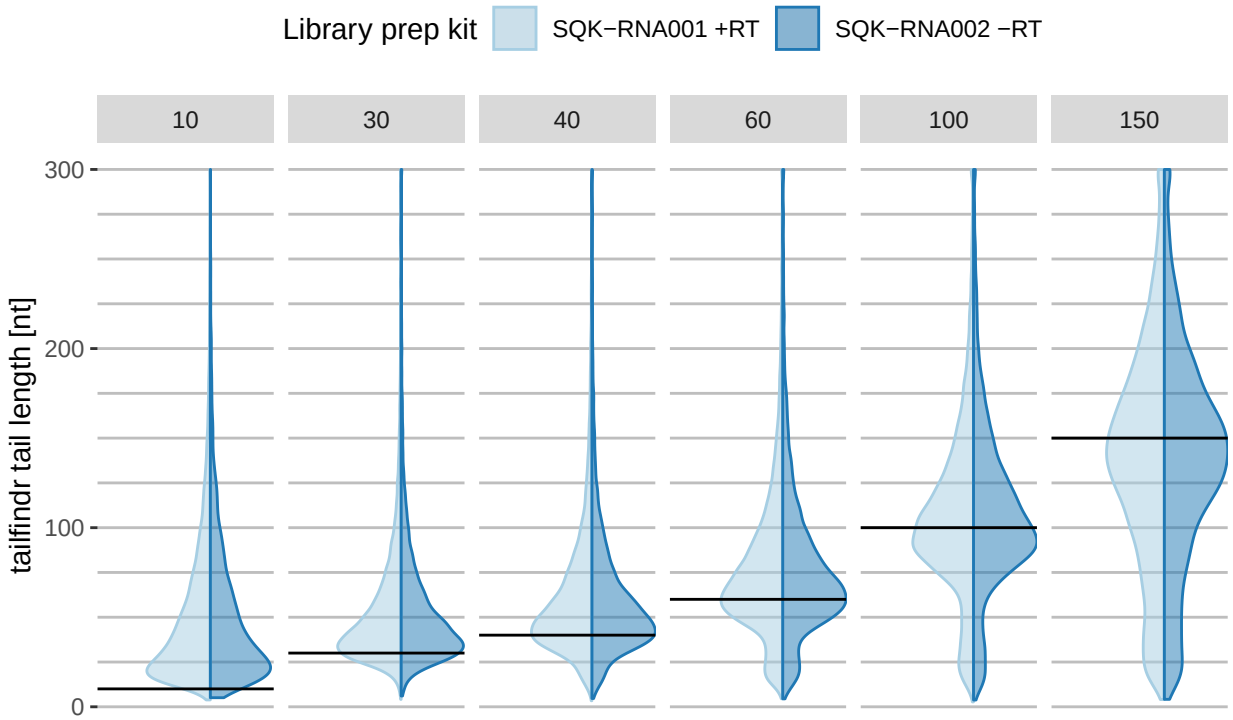
1. Comparing ONT Library prep conditions: SQK-RNA001 vs. SQK-RNA002

Create a dataset in which tail lengths are capped to 300 nt.

```
rna_kr_data_capped <- rna_kr_data %>%
  dplyr::mutate(tail_length_tf = ifelse(tail_length_tf >= 300, 300, tail_length_tf),
                tail_length_np = ifelse(tail_length_np >= 300, 300, tail_length_np))
```

Tail length densities

```
p <- ggplot(data = rna_kr_data_capped, aes(x = barcode, y = tail_length_tf,
                                           color = kit, fill = kit)) +
  geom_two_sided_flat_violin(position = position_nudge(x = .2, y = 0), alpha = .5) +
  facet_grid(~barcode, scales = 'free') +
  geom_hline(aes(yintercept=as.numeric(as.character(barcode))))
```



tailfindr tail length estimate is indepedent of sequencing kit used and whether or not reverse transcription step was performed. The black horizontal lines represent the expected poly(A) tail length.

Tail length statistics

```r
# define the function for computing standard error
std_err <- function(x) sd(x, na.rm = TRUE)/sqrt(length(x))

# summarize the data and display a table
summary_data <- rna_kr_data %>% group_by(barcode, kit) %>%
  summarise(read_count = n(),
            mean = mean(tail_length_tf, na.rm = TRUE),
            median = median(tail_length_tf, na.rm = TRUE),
            std_dev = sd(tail_length_tf, na.rm = TRUE),
            std_err = std_err(tail_length_tf))
summary_data %<>% mutate(cof_var = std_dev/mean)
kable(summary_data)
```

barcode	kit	read_count	mean	median	std_dev	std_err	cof_var
10	SQK-RNA001	43573	53.90961	40.55908	42.44215	0.2033240	0.7872837
10	SQK-RNA002	3463	52.98121	39.22500	42.78891	0.7271181	0.8076243
30	SQK-RNA001	36281	56.42209	44.88328	37.31151	0.1958859	0.6612927
30	SQK-RNA002	9356	56.52418	45.27000	38.54292	0.3984736	0.6818838
40	SQK-RNA001	12323	64.69623	53.24395	40.20295	0.3621594	0.6214111
40	SQK-RNA002	13994	62.12075	52.26000	37.20326	0.3144924	0.5988863
60	SQK-RNA001	44901	79.12513	69.63549	43.97424	0.2075249	0.5557556
60	SQK-RNA002	14690	78.73624	69.17000	43.95274	0.3626395	0.5582276
100	SQK-RNA001	26559	108.23113	102.27805	49.73756	0.3051959	0.4595495
100	SQK-RNA002	9831	109.32536	102.68000	49.09471	0.4951489	0.4490697
150	SQK-RNA001	21996	138.22862	139.41396	63.41088	0.4275550	0.4587392
150	SQK-RNA002	7271	138.42971	139.97000	61.02587	0.7156767	0.4408437

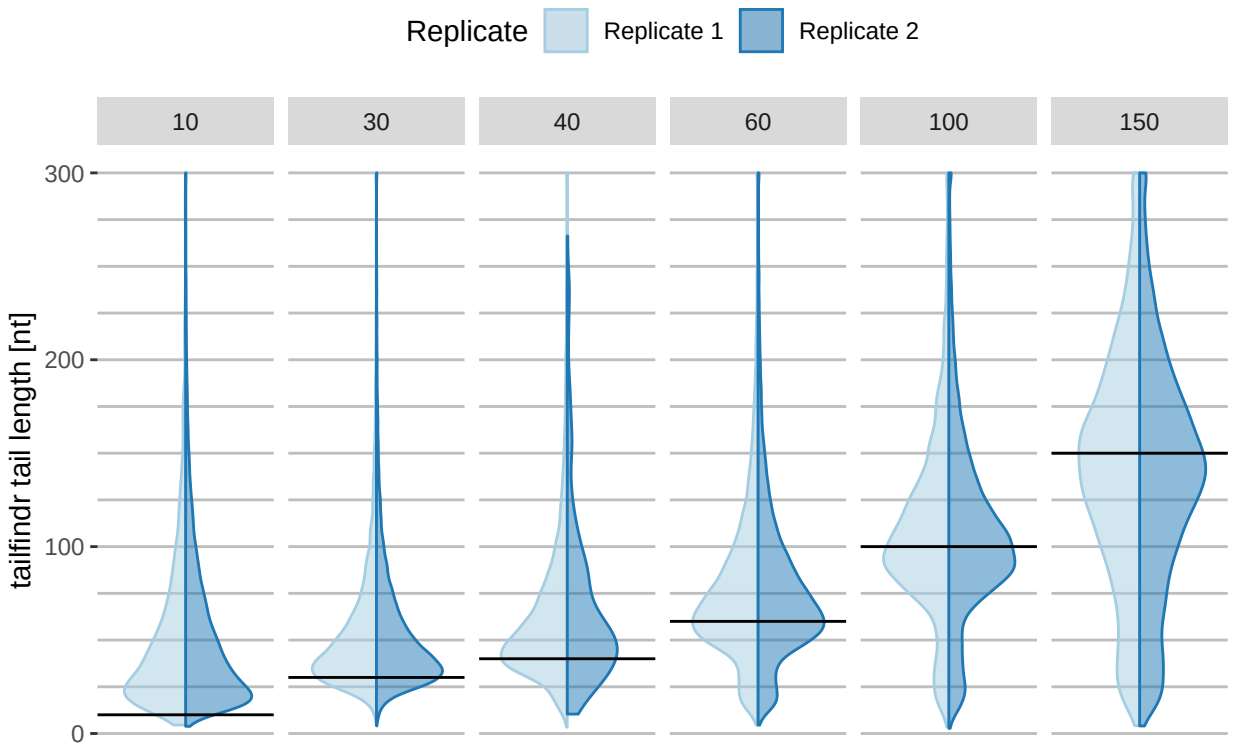
2. Comparing technical replicates: Replicate 1 vs Replicate 2 (both RNA001)

The first two replicates (both obtained with SQK-RNA001) are loaded from complete dataset.

```r
rna001_data <- rna_kr_data_capped %>% filter(replicate == 1 | replicate == 2)
```

Tail length densities

```r
p <- ggplot(rna001_data, aes(x = barcode, y = tail_length_tf,
                             color = replicate, fill = replicate)) +
  geom_two_sided_flat_violin(position = position_nudge(x = 0, y = 0), alpha = .5) +
  facet_grid(~barcode, scales = 'free') +
  geom_hline(aes(yintercept = as.numeric(as.character(barcode))))
```



tailfindr tail length estimate is robust across technical replicates.
The black horizontal lines represent the expected poly(A) tail length.

Tail length summaries

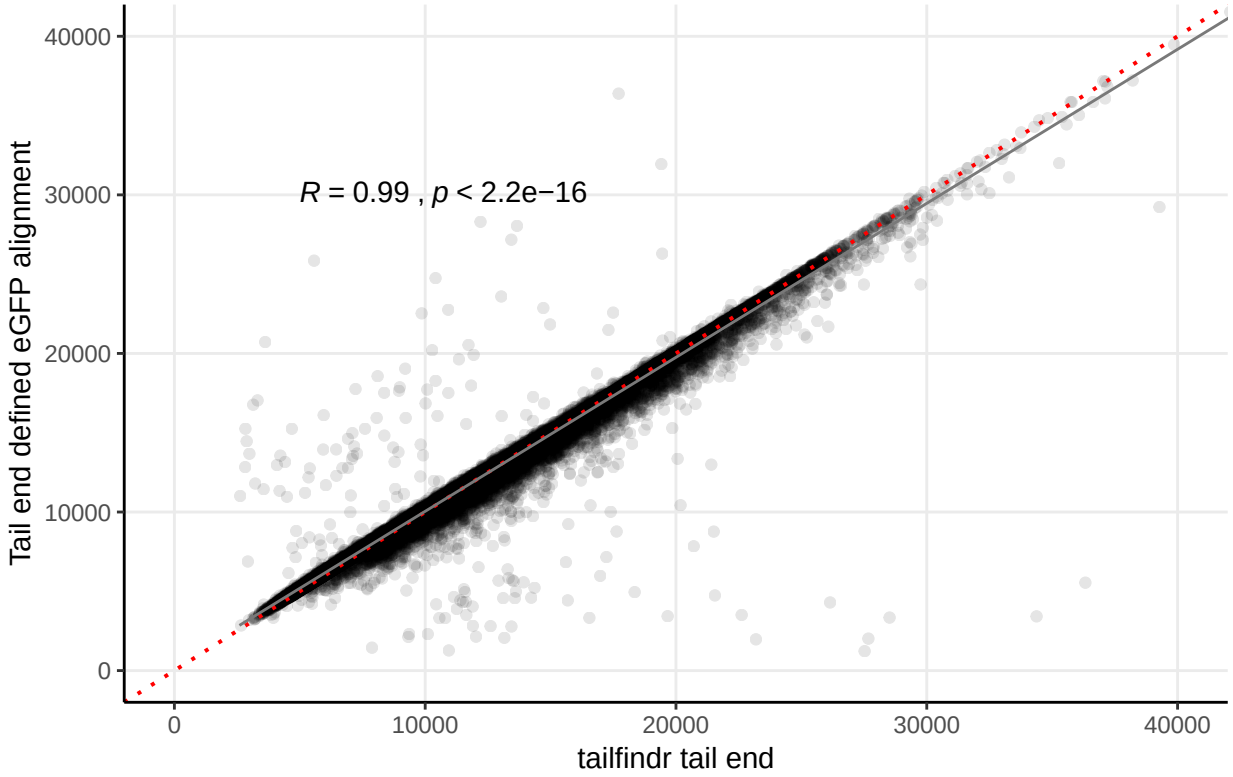
```
summary_data <- rna001_data %>% group_by(barcode, replicate) %>%
  summarise(read_count = n(),
            mean = mean(tail_length_tf, na.rm = TRUE),
            median = median(tail_length_tf, na.rm = TRUE),
            std_dev = sd(tail_length_tf, na.rm = TRUE),
            std_err = std_err(tail_length_tf))
summary_data %<>% mutate(cof_var = std_dev/mean)
kable(summary_data)
```

barcode	replicate	read_count	mean	median	std_dev	std_err	cof_var
10	1	3148	54.30944	40.36212	44.34128	0.7902979	0.8164562
10	2	40425	53.85883	40.59769	42.16667	0.2097222	0.7829110
30	1	7724	56.71610	45.33672	37.85945	0.4307777	0.6675257
30	2	28557	56.30455	44.79311	36.88542	0.2182722	0.6551055
40	1	12044	64.57766	53.23498	39.87166	0.3633113	0.6174218
40	2	279	68.75501	53.55738	46.98745	2.8130653	0.6834041
60	1	11679	79.41593	69.51284	44.37602	0.4106252	0.5587799
60	2	33222	78.95817	69.68956	43.46967	0.2384918	0.5505405
100	1	5439	110.12318	103.35213	50.08909	0.6791780	0.4548460
100	2	21120	107.60240	101.95574	49.00852	0.3372288	0.4554593
150	1	5219	140.79093	141.94923	63.12239	0.8737553	0.4483413
150	2	16777	137.07325	138.76908	62.40588	0.4818017	0.4552739

3. tailfindr poly(A) end vs. sequence end by alignment of eGFP

To test whether poly(A) end as defined by tailfindr and the sequence end of eGFP as defined by sequence alignment match, a scatter plot is generated:

```
p <- ggplot(rna_kr_data, aes(x = tail_end_tf, y = transcript_alignment_start)) +
    geom_point(shape = 21, colour = 'black', fill = 'black',
               size = 2, stroke=0, alpha = 0.1) +
    geom_abline(intercept = 0, slope = 1, color="red",
                linetype = 'dotted', size = 0.7) +
    geom_smooth(method = 'lm', formula = y~x,
                color="#797979", fullrange = TRUE, se = FALSE, size = 0.5) +
    stat_cor(method = "pearson", label.x = 5000, label.y = 30000) +
    coord_cartesian(xlim = c(0, 40000), ylim = c(0, 40000))
```



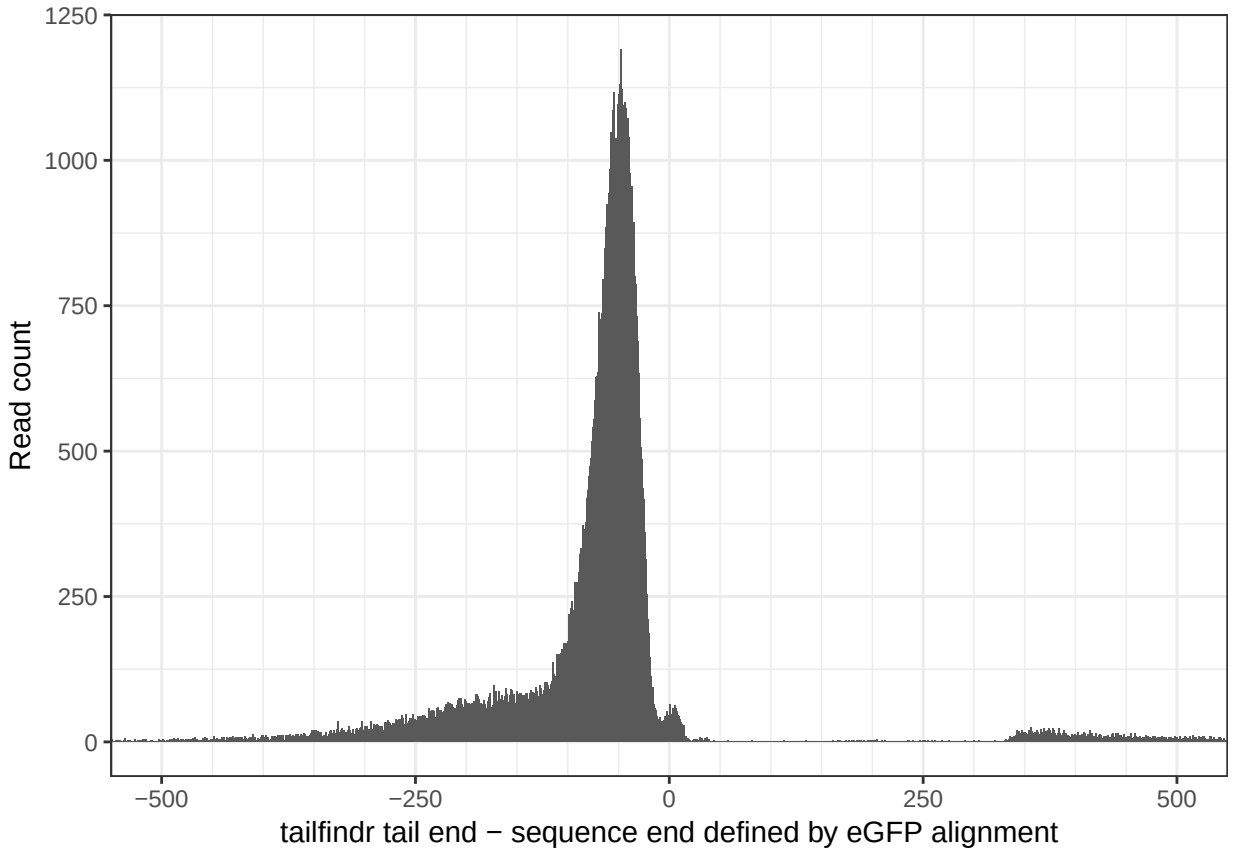
The red dotted line represents x = y.
The grey line represents the linear model fit.

To better visualize the precision of end definition, a histogram of the difference between tailfindr poly(A) end and the sequence alignment position is generated:

```
data <- mutate(rna_kr_data, diff = tail_end_tf - transcript_alignment_start)
p <- ggplot(data, aes(x = diff)) +
  geom_histogram(binwidth = 1)
```

```
p <- p +
  theme_bw() +
  coord_cartesian(xlim = c(-500, 500)) +
  scale_x_continuous(minor_breaks = seq(-500, 500, 50)) +
  xlab('tailfindr tail end - sequence end defined by eGFP alignment') +
  ylab('Read count')
```

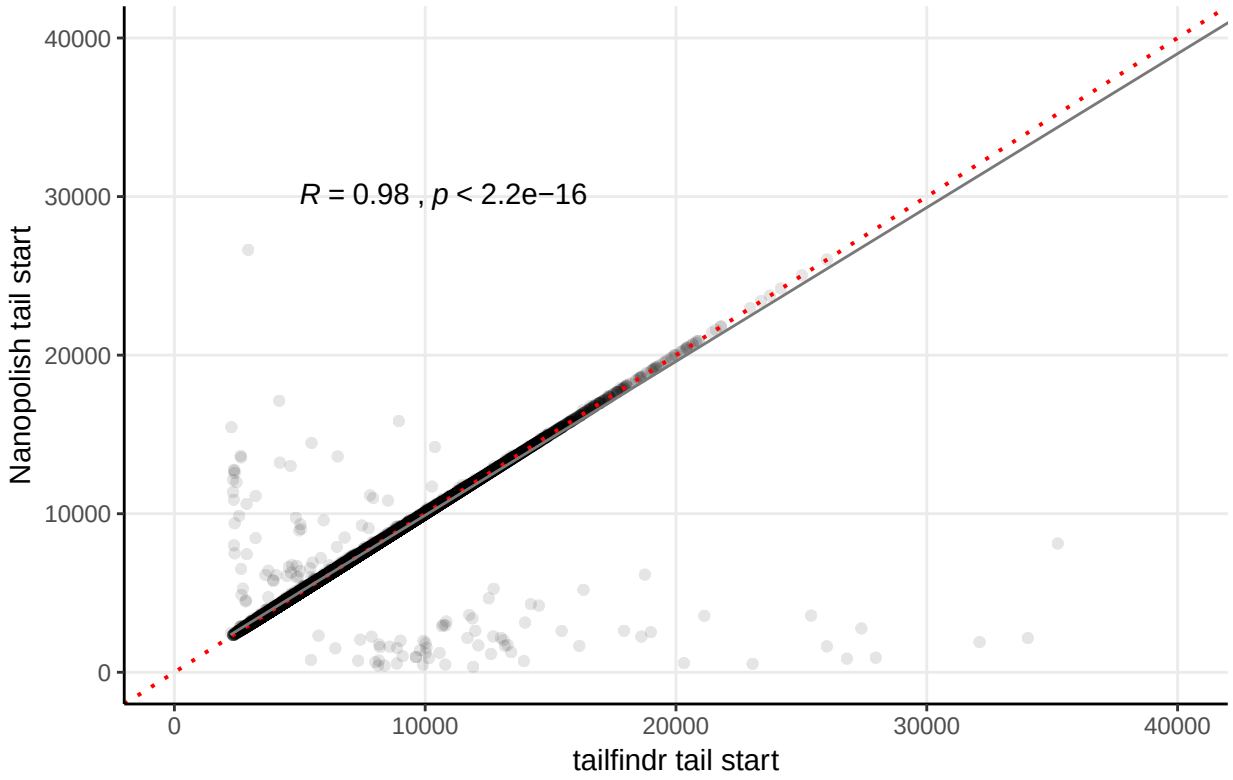
```
p
```



4. Nanopolish tail start estimate vs. tailfindr tail start estimate

Let us generate a scatter plot of tailfindr and Nanopolish poly(A) tail start estimates:

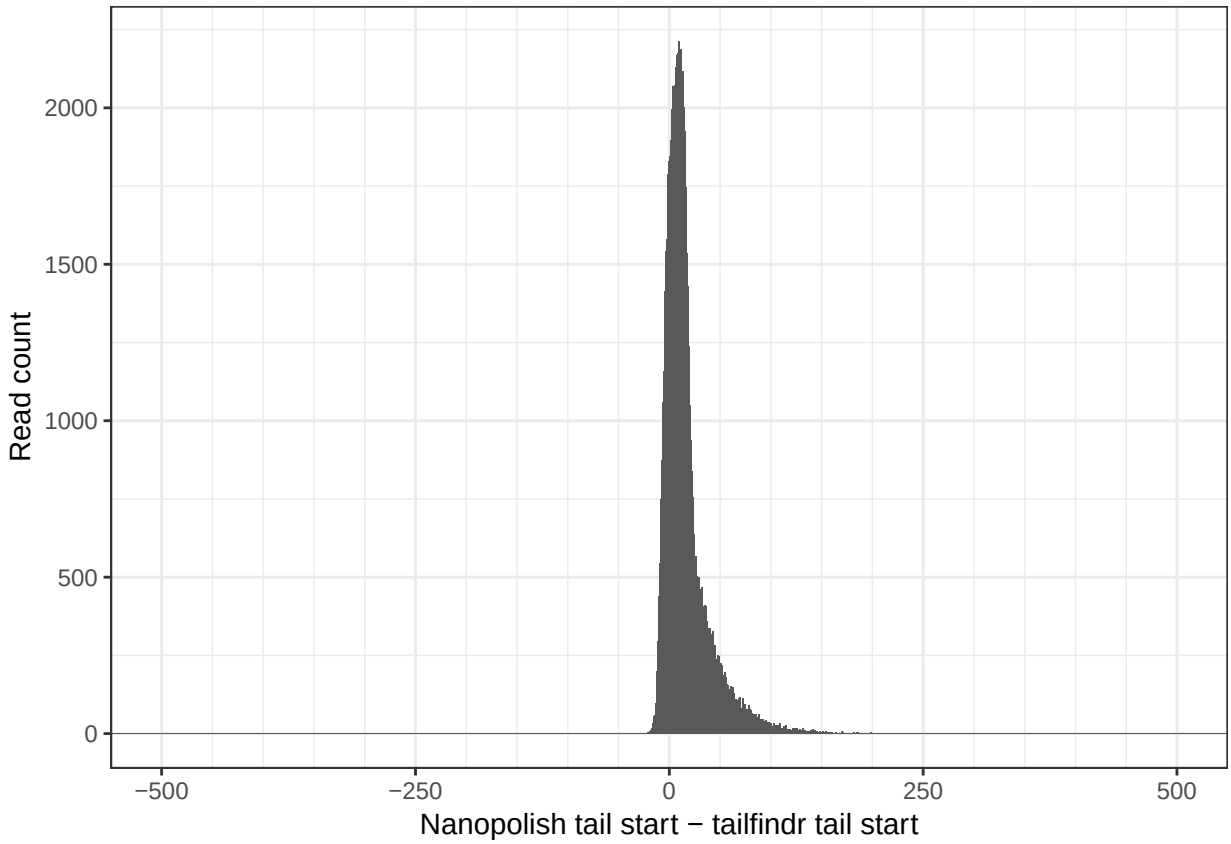
```
p <- ggplot(rna_kr_data, aes(x = tail_start_tf, y = tail_start_np)) +
    geom_point(shape = 21, colour = 'black', fill = 'black',
               size = 2, stroke=0, alpha = 0.1) +
    geom_abline(intercept = 0, slope = 1, color="red",
                linetype = 'dotted', size = 0.7) +
    geom_smooth(method = 'lm', formula = y~x,
                color="#797979", fullrange = TRUE, se = FALSE, size = 0.5) +
    stat_cor(method = "pearson", label.x = 5000, label.y = 30000) +
    coord_cartesian(xlim = c(0, 40000), ylim = c(0, 40000))
```



The red dotted line represents x = y.
The grey line represents the linear model fit.

To better visualize the overlap in poly(A) start detection, a histogram of the difference between Nanopolish and tailfindr poly(A) start is generated:

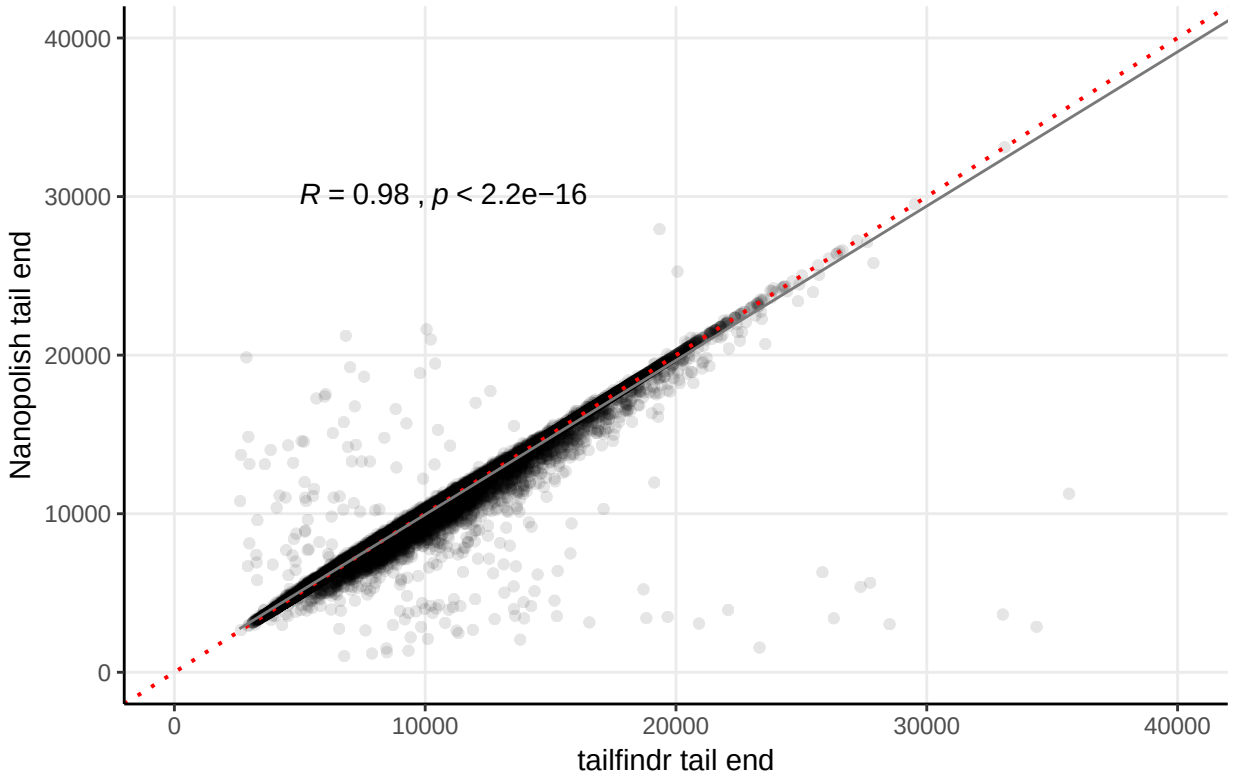
```
data <- mutate(rna_kr_data, diff = tail_start_np - tail_start_tf)
p <- ggplot(data, aes(x = diff)) +
  geom_histogram(binwidth = 1) +
  theme_bw() +
  coord_cartesian(xlim = c(-500, 500)) +
  scale_x_continuous(minor_breaks = seq(-500, 500, 50)) +
  xlab('Nanopolish tail start - tailfindr tail start') +
  ylab('Read count')
p
```



5. Nanopolish tail end estimate vs. tailfindr tail end estimate

Let us generate a scatter plot of tailfindr and Nanopolish poly(A) tail end estimates:

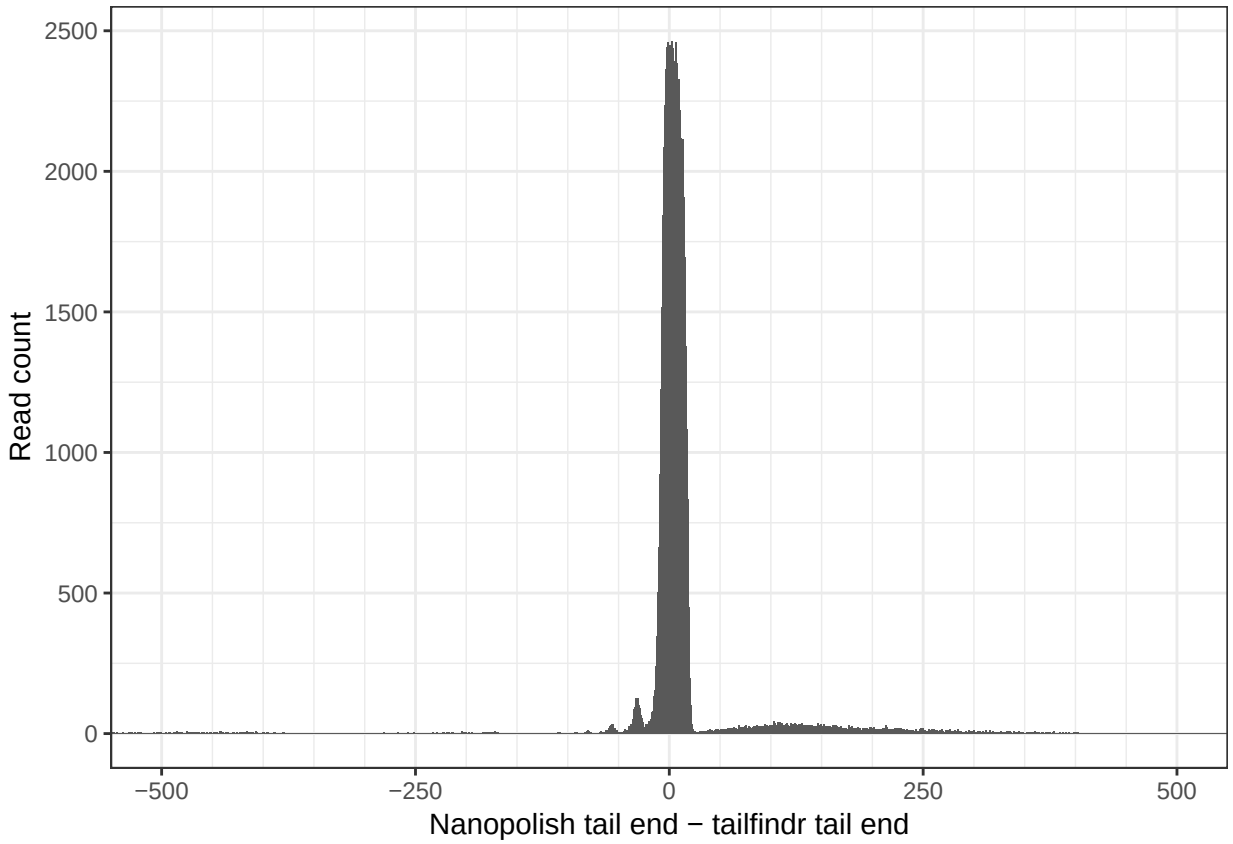
```
p <- ggplot(rna_kr_data, aes(x = tail_end_tf, y = tail_end_np)) +
    geom_point(shape = 21, colour = 'black', fill = 'black',
               size = 2, stroke=0, alpha = 0.1) +
    geom_abline(intercept = 0, slope = 1, color="red",
                linetype = 'dotted', size = 0.7) +
    geom_smooth(method = 'lm', formula = y~x,
                color="#797979", fullrange = TRUE, se = FALSE, size = 0.5) +
    stat_cor(method = "pearson", label.x = 5000, label.y = 30000) +
    coord_cartesian(xlim = c(0, 40000), ylim = c(0, 40000))
```



The red dotted line represents x = y.
The grey line represents the linear model fit.

To better visualize the overlap in poly(A) end detection, a histogram of the difference between Nanopolish and tailfindr poly(A) end is generated:

```
data <- mutate(rna_kr_data, diff = tail_end_np - tail_end_tf)
p <- ggplot(data, aes(x = diff)) +
  geom_histogram(binwidth = 1) +
  theme_bw() +
  coord_cartesian(xlim = c(-500, 500)) +
  scale_x_continuous(minor_breaks = seq(-500, 500, 50)) +
  xlab('Nanopolish tail end - tailfindr tail end') +
  ylab('Read count')
p
```


2500
2000
1500
1000
500
0
Read count
−500
−250
0
250
500
Nanopolish tail end − tailfindr tail end