parallel-linear

2025-10-02

Fitting linear model for the quicksort data

```
library(tidyr)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(nlstools)
```

```
##
## 'nlstools' has been loaded.
```

```
## IMPORTANT NOTICE: Most nonlinear regression models and data set examples
```

```
## related to predictive microbiolgy have been moved to the package 'nlsMicrobio'
```

```
library(purrr)
library(broom)
```

```
data <- read.csv("measurments.csv")
head(data)
```

```
##   Size Seq     Par Libc
## 1    1   0 4.6e-05    0
## 2    1   0 4.2e-05    0
## 3    1   0 3.7e-05    0
## 4    1   0 4.4e-05    0
## 5    1   0 4.0e-05    0
## 6    1   0 3.9e-05    0
```

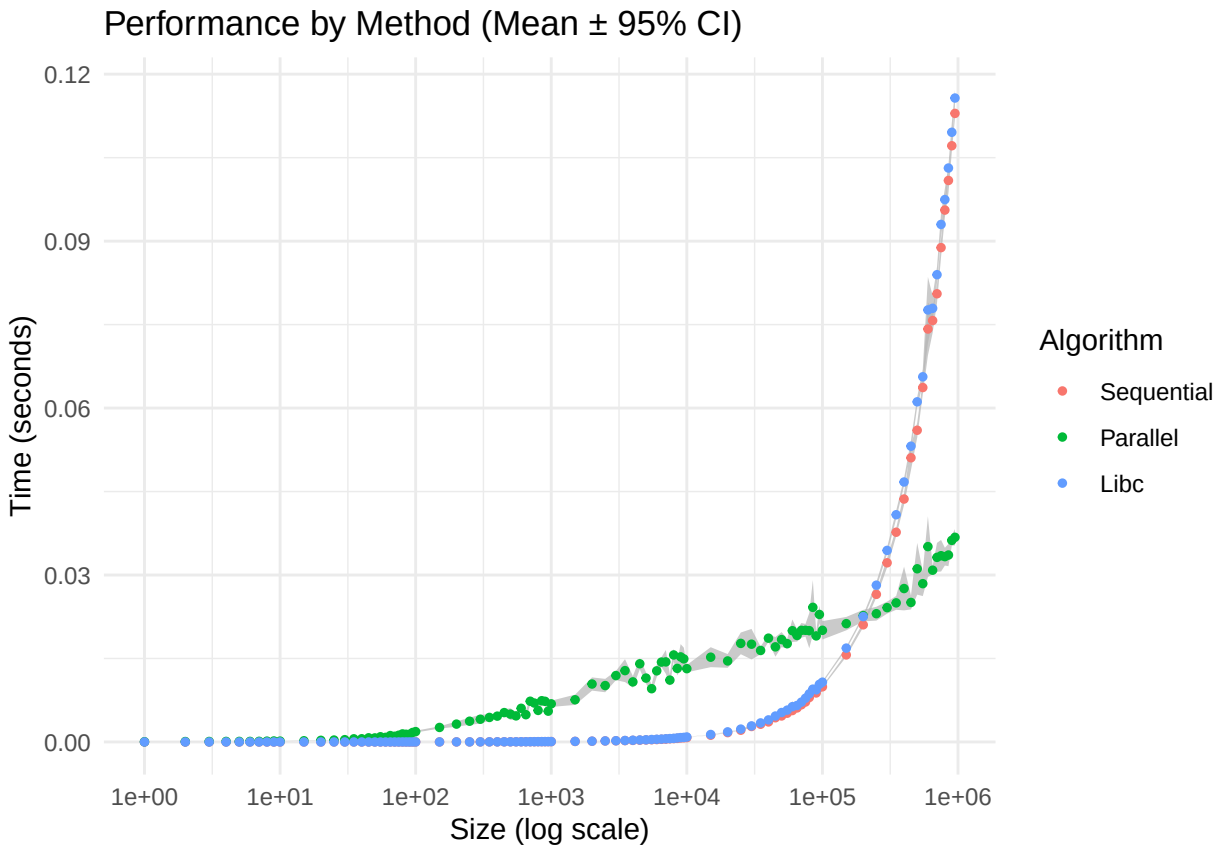
```
data_long <- pivot_longer(data, cols = c("Seq", "Par", "Libc"),
                          names_to = "Method", values_to = "Time")
head(data_long)
```

```
## # A tibble: 6 x 3
##    Size Method     Time
##   <int> <chr>     <dbl>
## 1     1 Seq    0
## 2     1 Par    0.000046
```

```
## 3     1 Libc   0
## 4     1 Seq    0
## 5     1 Par    0.000042
## 6     1 Libc   0
```

Visualization means of time for the sizes, on logarithmic scale

```r
ggplot(data_long, aes(x = Size, y = Time, color = Method)) +
  stat_summary(
    aes(group = Method),
    fun.data = mean_cl_normal,
    geom = "ribbon",
    fill = "grey70",
    alpha = 0.7,
    color = NA
  ) +
  # Mean points
  stat_summary(
    fun = mean,
    geom = "point",
    size = 1
  ) +
  scale_x_log10(
    breaks = c(1, 10, 100, 1000, 10000, 100000, 1000000)
  ) +
  scale_color_discrete(
    name = "Algorithm",
    labels = c("Sequential", "Parallel", "Libc")
  ) +
  labs(
    title = "Performance by Method (Mean ± 95% CI)",
    x = "Size (log scale)",
    y = "Time (seconds)"
  ) +
  theme_minimal()
```



Fitting linear model y ~ a + b(nlogn) for every algorithm

```
library(dplyr)


data_mean <- data_long %>%
  group_by(Method, Size) %>%
  summarise(MeanTime = mean(Time), .groups = "drop") %>%
  mutate(nlogn = Size * log(Size))


models <- data_mean %>%
  group_by(Method) %>%
  group_map(~ lm(MeanTime ~ nlogn, data = .x)) %>%
  setNames(unique(data_mean$Method))
```

Summary of Libc regression

```
summary(models[["Libc"]])
```

```
##
## Call:
## lm(formula = MeanTime ~ nlogn, data = .x)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
```

```
## -0.0014963 -0.0000080  0.0000147  0.0000169  0.0046949
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.459e-05  6.366e-05  -0.229    0.819
## nlogn        8.709e-09  1.888e-11 461.418   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0005777 on 97 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9995
## F-statistic: 2.129e+05 on 1 and 97 DF,  p-value: < 2.2e-16
```

Summary of Parallel regression

```
summary(models[["Par"]])
```

```
##
## Call:
## lm(formula = MeanTime ~ nlogn, data = .x)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0080665 -0.0067312 -0.0008216  0.0060013  0.0134463
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.108e-03  7.166e-04   11.31   <2e-16 ***
## nlogn       2.732e-09  2.124e-10   12.86   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.006502 on 97 degrees of freedom
## Multiple R-squared:  0.6303, Adjusted R-squared:  0.6265
## F-statistic: 165.4 on 1 and 97 DF,  p-value: < 2.2e-16
```

Summary of Sequential regression

```
summary(models[["Seq"]])
```

```
##
## Call:
## lm(formula = MeanTime ~ nlogn, data = .x)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0021237 -0.0001152 -0.0001108 -0.0000290  0.0056550
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.162e-04  8.067e-05    1.44    0.153
## nlogn       9.003e-09  2.392e-11  376.43   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.0007319 on 97 degrees of freedom
## Multiple R-squared:  0.9993, Adjusted R-squared:  0.9993
## F-statistic: 1.417e+05 on 1 and 97 DF,  p-value: < 2.2e-16
```

```
coefs <- imap_dfr(
  models,
  ~ data.frame(
      Method = .y,
      Intercept = coef(.x)[1],
      Slope = coef(.x)[2]
    )
)

data_fit <- data_mean %>%
  left_join(coefs, by = "Method") %>%
  mutate(Fitted = Intercept + Slope * nlogn)
```

Visualization of regression, qqplots, residuals

```
library(ggplot2)
library(dplyr)
library(scales)
```

```
##
## Attaching package: 'scales'
```

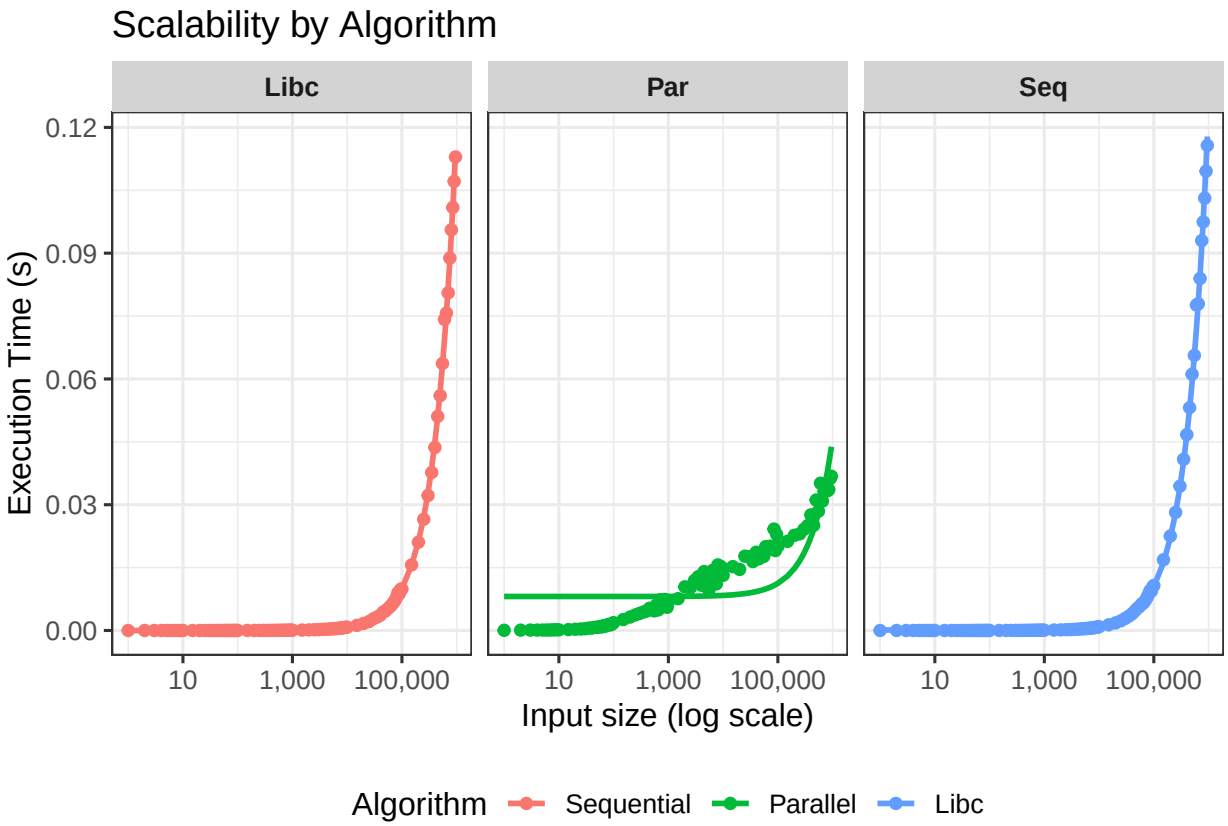
```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
ggplot(data_mean, aes(x = Size, y = MeanTime, color = Method)) +
  geom_point(linewidth = 1) +
  geom_line(data = data_fit, aes(x = Size, y = Fitted, color = Method), size = 1) +
  scale_color_discrete(
    name = "Algorithm",
    labels = c("Sequential", "Parallel", "Libc")
  ) +
  facet_wrap(~Method) +
  scale_x_log10(labels = comma) +
  labs(
    title = "Scalability by Algorithm",
    x = "Input size (log scale)",
    y = "Execution Time (s)",
    color = "Algorithm"
  ) +
  theme_bw(base_size = 12) +
  theme(
    legend.position = "bottom",
    strip.background = element_rect(fill = "lightgray", color = NA),
    strip.text = element_text(face = "bold")
  )
```

```
## Warning in geom_point(linewidth = 1): Ignoring unknown parameters: `linewidth`
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
```

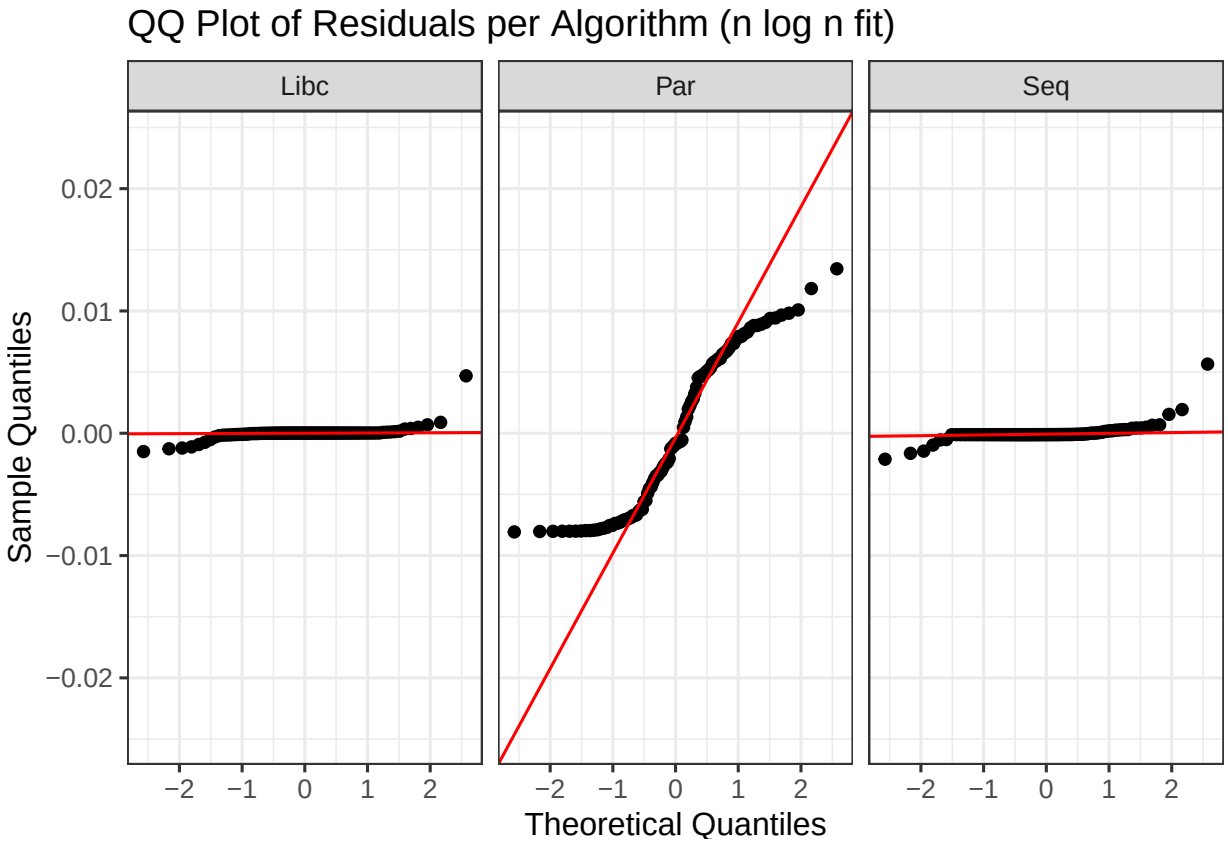
```
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```r
library(dplyr)
library(ggplot2)
library(tidyr)

data_resid <- data_mean %>%
  group_by(Method) %>%
  group_modify(~ {
    mod <- lm(MeanTime ~ nlogn, data = .x)
    .x$residuals <- resid(mod)
    .x
  })
```

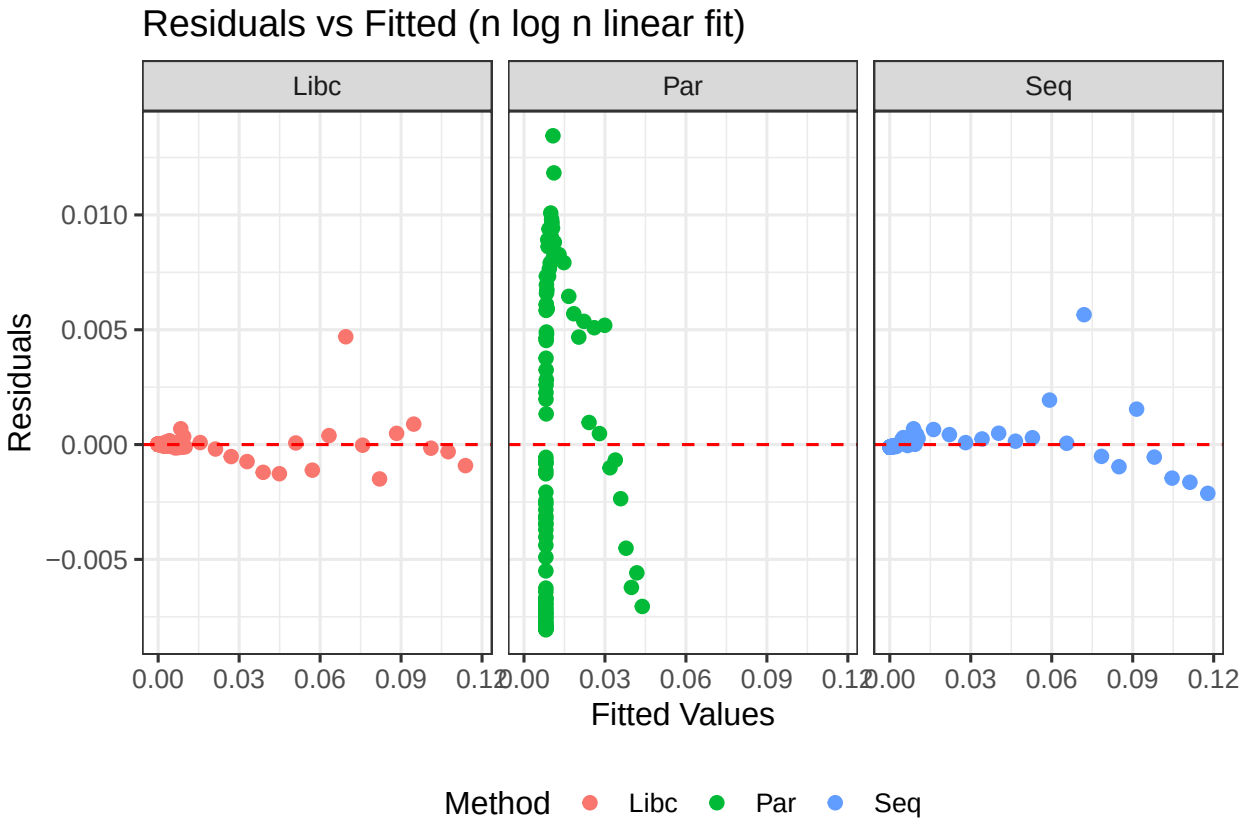
```r
ggplot(data_resid, aes(sample = residuals)) +
  stat_qq() +
  stat_qq_line(color = "red") +
  facet_wrap(~Method) +
  labs(
    title = "QQ Plot of Residuals per Algorithm (n log n fit)",
    x = "Theoretical Quantiles",
    y = "Sample Quantiles"
  ) +
  theme_bw(base_size = 12)
```




```r
library(dplyr)
library(ggplot2)

data_resid <- data_mean %>%
  group_by(Method) %>%
  group_modify(~ {
    mod <- lm(MeanTime ~ nlogn, data = .x)
    .x <- .x %>%
      mutate(
        Fitted = predict(mod, newdata = .x),
        Residuals = residuals(mod)
      )
    .x
  })
```

```r
ggplot(data_resid, aes(x = Fitted, y = Residuals, color = Method)) +
  geom_point(size = 2) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  facet_wrap(~Method) +
  labs(
    title = "Residuals vs Fitted (n log n linear fit)",
    x = "Fitted Values",
    y = "Residuals"
  ) +
  theme_bw(base_size = 12) +
  theme(legend.position = "bottom")
```



Evaluation of the fit

Based on the summary and visualization of regression curves, residuals, and QQ plots for residuals, the chosen linear models seem to fit the data correctly for the Sequential and Libc implementations of parallel sort. Residual values for those implementations seem low. The residual standard error is also low, suggesting low variance. The R^2 for those implementations is high, which means that the variance of the model explains almost all of the total variance. QQ plots for residuals suggest that the error in this case has a normal distribution, and residuals vs fitted plots show that residual values are mostly consistent and fall around zero.

The fit is not as good for the parallel implementation, which is visible in the plots showing the regression curve. The residual median and quantiles seem similar to those of the other implementations, but an R^2 value around 0.6 shows that the total variance is not properly explained by the model. QQ plots and visualization of residuals also suggest that the model is not a good fit, as the error does not seem to follow a normal distribution. A different model might be a better fit in this case.