Numerical instability

Dennis Scheper

2024-06-29

Deliverable 2

Under `data/` you will find all the necessary CSVs with the average PHRED score per base position. We utilized 1-4 workers and repeated the process 3 times for worker configuration, which resulted in 12 different CSVs. This notebook will begin by presenting the the results and thereafter discuss their implications. Timing of the processes are present under `data/` too.

To start, we will combine the CSV files into a single dataframe.

```r
library(dplyr)
library(stringr)
library(janitor)
library(tidyr)
library(ggplot2)
library(glue)

path <- setwd("~/Desktop/deliverable2/data")

file_list <- list.files(path = path, recursive = TRUE,
                        full.names = TRUE,pattern = "\\.csv$")

read <- function(csv_file) {
  base <- basename(csv_file) %>% # get the basename
    str_extract("(?<=output_)\\d+_\\d+")

  end <- csv_file %>%
    read.csv(., sep=",", quote='') %>%
    dplyr::select(-1) %>%
    rename_with(~base, .cols=1) # rename the columns to basename
  return(end)
}

data_list <- lapply(file_list, read)
all_data <- bind_cols(data_list) %>%
  mutate(position = seq(1,100,1)) %>% # 1-100 base positions
  select(last_col(), everything()) %>% t() %>%
  as.data.frame() %>%
  row_to_names(1)
```

Next, we will answer the following questions:

- How much variation in PHRED scores do you observe within a run?

- Does this appear to be constant over the read?
- How much variation do you observe in PHRED scores between runs with fewer or more workers?
- For how many workers is the standard deviation the largest?

Below, we start by assigning a worker ID and run ID to keep track of each process. Next, we calculate the standard deviation for each run on a row-wise basis. In Figure 1, we display the standard deviation per run for each worker. It is evident that the highest standard deviation occurs with a single worker and tends to decrease as the number of workers increases. Furthermore, the standard deviation appears to decrease slightly as the number of runs increases.

```r
all_data <- all_data %>%
  mutate(n_workers=rep(1:4, each=3,length.out=nrow(all_data))) %>%
  mutate(n_run=rep(1:3, length.out=nrow(all_data))) %>%
  rowwise() %>%
  mutate(SD = sd(c_across(everything()))) # go over rows column wise

ggplot(all_data, aes(x = n_run, y = SD, color = as.factor(n_workers))) +
  geom_point() +
  geom_line() +
  labs(title="Standard deviation of PHRED scores per run",  x="Amount of runs") +
  scale_color_discrete(name = "Number of workers")
```

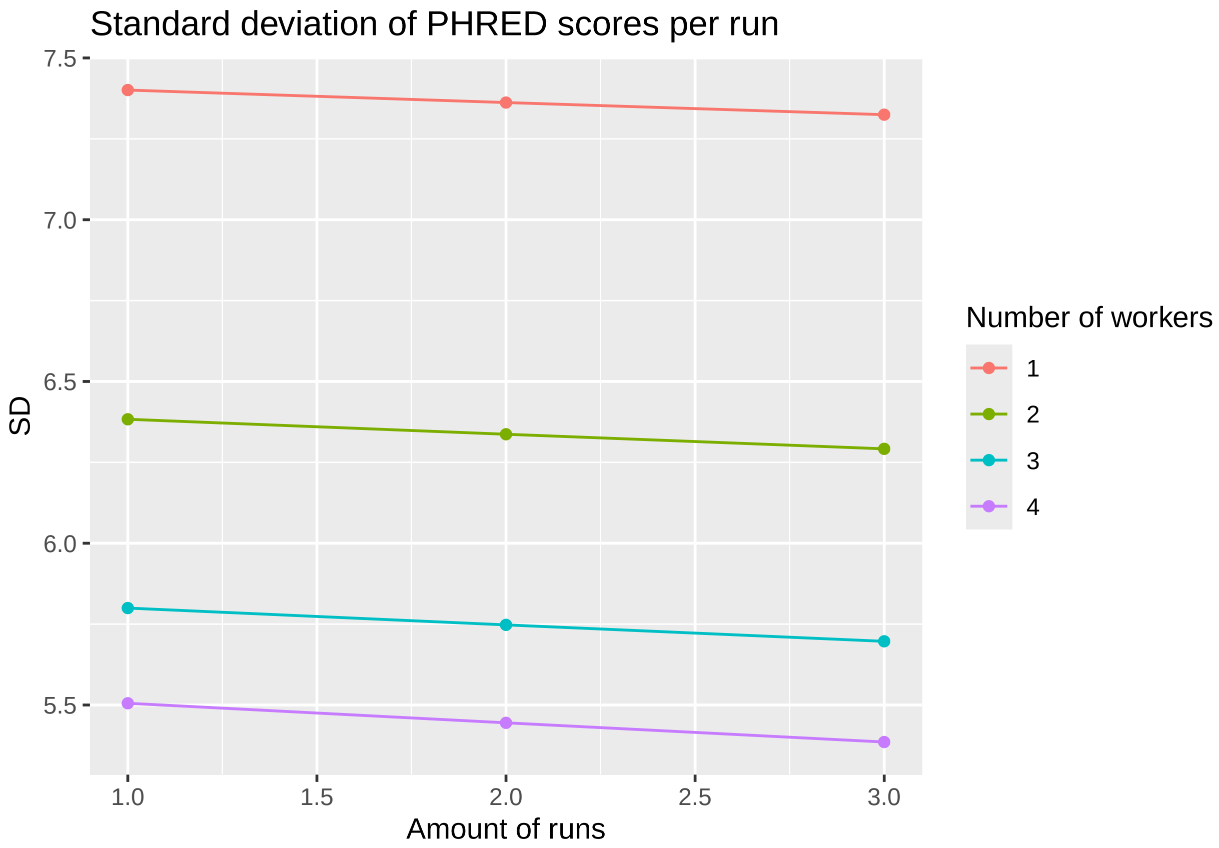

Figure 1: Standard deviation between PHRED scores per run and visualized by the amount of workers. The standard deviation seems to go down when the amount of workers increases.

When we plot the number of workers against the standard deviation in PHRED scores, Figure 2 presents a similar trend to Figure 1. We first grouped the data by worker ID and calculated the mean standard deviation. Figure 2 illustrates that the average standard deviation decreases from around 7.5 with one worker to below 5.5 with four workers. Consequently, the highest standard deviation is observed when only one worker is utilized.

```r
sd_by_worker <- all_data %>%
  group_by(n_workers) %>%
  summarise(mean_sd = mean(SD))

ggplot(sd_by_worker, aes(x = n_workers, y = mean_sd)) +
  geom_line() +
  geom_point() +
  labs(title = "Average SD per worker", x = "Number of workers", y = "Average SD")
```

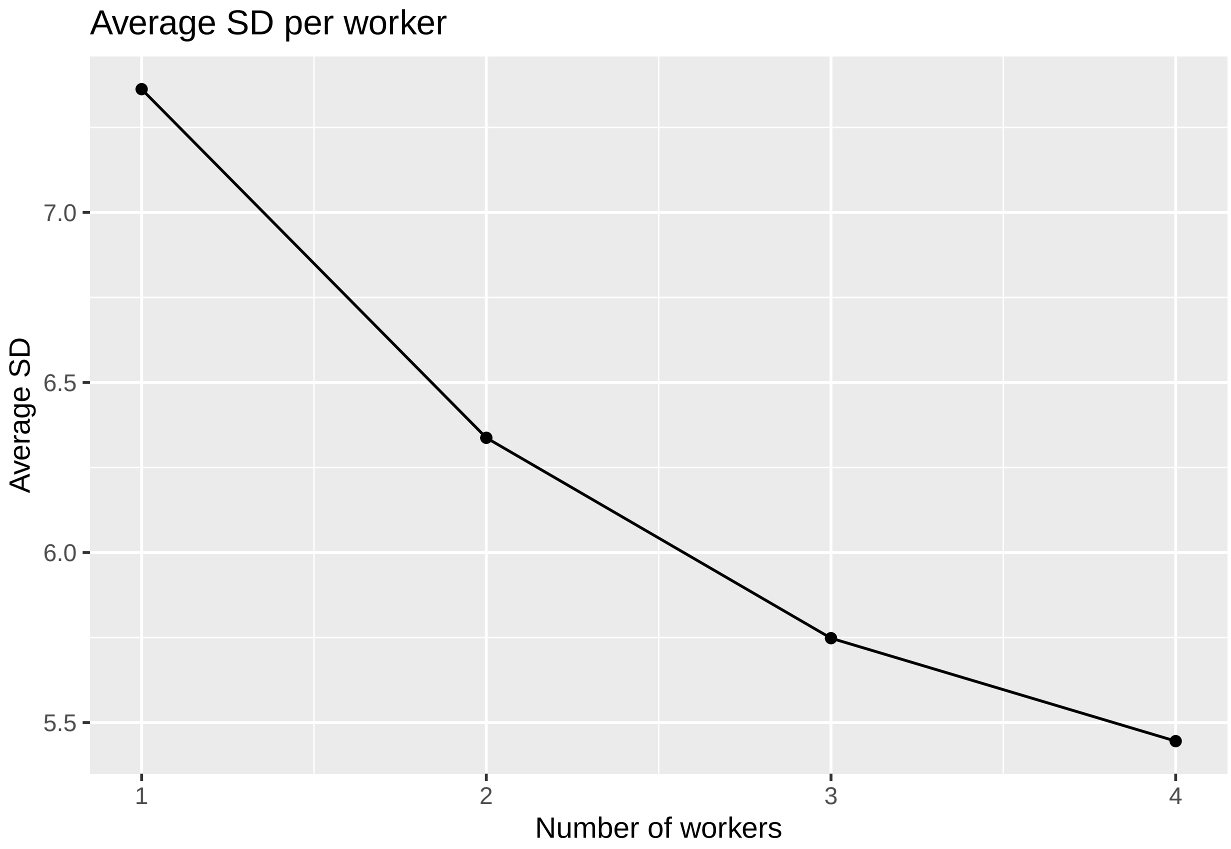

Figure 2: Mean standard deviation per worker.

```r
max_sd_worker <- sd_by_worker %>%
  filter(mean_sd == max(mean_sd)) %>%
  pull(n_workers)

cat(glue("Amount of workers with the highest mean SD
         ({round(max(sd_by_worker$mean_sd), 4)}): {max_sd_worker}"))
```

```
## Amount of workers with the highest mean SD
## (7.3625): 1
```

The results suggest that the amount of workers let the standard deviation decrease. This might be since more workers can distribute the computational load more evenly. Thereby, the likelihood of variations caused by an individual worker is reduced, which we saw are present in the amount of runs for the same worker. Here, even though the process is the same, the standard deviation still fluctuated. These slight fluctuations between runs are caused by *floating-point calculations*. Consequently, using too many workers can have a detrimental effect too, as it creates variations in the results. Therefore, balancing the number of workers with the amount of data is an important consideration to ensure optimal performance and minimal variability.