Additional information to the preregistration for ‘Data Processing Strategies to Determine Maximum Oxygen Uptake: A Systematic Scoping Review and Experimental Comparison’

Simon Nolte

Part II: Experimental Comparison

This document includes the additional information accompanying the preregistration listed in the file `preregistration.pdf`. It includes example scripts and detailed workflows of the planed analysis steps.

```
# Packages used for the data workflow

library(purrr)
library(spiro)
library(tidyr)
library(ggplot2)
library(scales)
library(MetBrewer)
```

Data Source

A total of n = 76 data sets will be used for analysis. The exercise tests were conducted during two studies and had the exact same methodology (Quittmann et al., under review, unpubl.) . All individual data files will be saved in a `data` folder.

```
files <- list.files("data/", full.names = TRUE)
```

Using different processing strategies on the data

The most common data processing strategies will be tested on the data for a wide range of parameter inputs. This will happen using the `spiro` package (Nolte, 2022).

I tested this exemplary code for two data files unrelated to the work presented here (two constant load tests) and for the exemplary strategies of breath- and time-based moving averages for the parameter range 5-60.

The resulting matrix (rows: data files, columns: [processing strategy_parameter]) will be saved as an `csv` file.

```
vo2max <- function(parameter, method, data) {
  if (method == "ba") {
    parameter_exp <- paste0(parameter, "b")
  } else {
    parameter_exp <- parameter
```

```r
  }
  max_values <- spiro_max(data, parameter_exp)
  out <- data.frame(
    name = max_values$VO2_rel
  )
  colnames(out) <- paste(method, parameter, sep = "_")
  out
}

# level 1: iterate over each data file
results <- purrr::map_dfr(
  .x = files,
  .f = function(file) {
    # level 2: iterate over each processing method
    data <- spiro(file)
    purrr::map_dfc(
      .x = c("ta", "ba"),
      .f = function(method, data) {
        # level 3: iterate over each parameter range
        purrr::map_dfc(
          .x = seq.int(5,60),
          .f = vo2max,
          method = method,
          data = data
        )
      },
      data = data
    )
  }
)

write.csv(results, "results.csv")
```

Normalize results

I now normalize the results to a reference procedure (the most common according to the review). For illustration and code testing, I here use the 10 second breath-averaging as the reference procedure.

```r
results_norm <- results / results$ba_10
results_norm$id <- as.factor(seq_len(nrow(results_norm)))
```

Plot data

I will plot the effect of data averaging strategy for varying parameter on $\dot{V}O_{2max}$ values.

Turn data in tidy format

To plot the data using `ggplot` I have to bring it into tidy format first.

```r
results_tidy <- pivot_longer(
  data = results_norm,
  cols = !id,
  names_to = c("type", "parameter"),
  names_pattern = "(.*)_(.*)",
  names_transform = list(
    parameter = ~ as.numeric(.x),
    type = ~ factor(.x,
      levels = c("ta", "ba"),
      labels = c("time average", "breath average")
    )
  )
)

# calculate normalized results as percent difference
per_diff <- (results_tidy$value - 1) / 1
```

Create plot

This is an exemplary plot to illustrate the supposed concept of visualisation. The final plot will rather display the median line with the 2.5 and 97.5 quantiles. Other processing strategies involving parameters that cannot be displayed on the given x-scale may be incorporated into the plot using appropriate techniques.

```
ggplot(results_tidy, aes(x = parameter, y = per_diff, colour = type, fill = type)) +
  geom_hline(
    yintercept = 0,
    colour = "grey",
    alpha = 0.15
  ) +
  stat_summary(geom = "line", fun = "mean") +
  stat_summary(
    geom = "ribbon",
    fun.min = "min",
    fun.max = "max",
    alpha = 0.15,
    colour = NA
  ) +
  scale_x_continuous(
    name = "Parameter",
    breaks = c(15, 30, 45, 60)
  ) +
  scale_y_continuous(
    name = "% to Reference",
    labels = scales::percent,
    limits = c(-0.07,0.1)
  ) +
  scale_color_manual(
    values = met.brewer("OKeeffe1", 2)
  ) +
  scale_fill_manual(
    values = met.brewer("OKeeffe1", 2)
  ) +
  theme_classic() +
  theme(
    legend.position = c(0.8,0.8)
  )

ggsave("example_plot.png", width = 6, height = 4, dpi = 300, bg = "white")
```

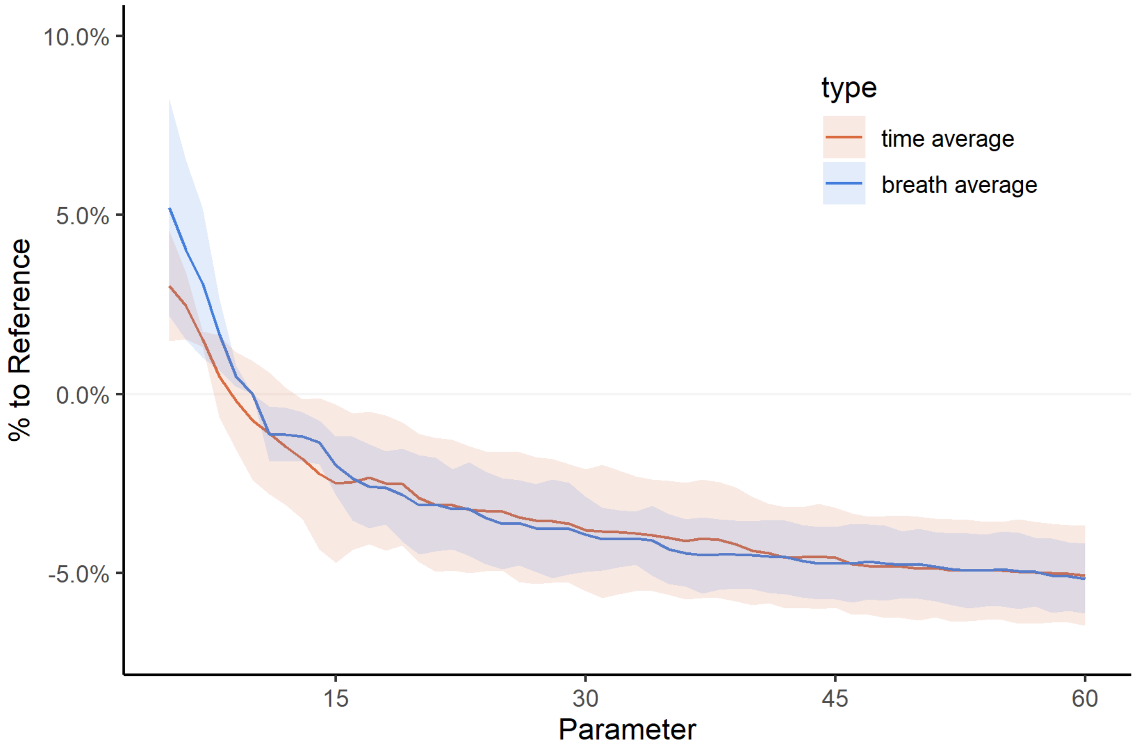

Figure 1: Proof of concept for a plot that illustrates differences between different data processing strategies

References

Nolte, S. (2022). *Spiro: Manage data from cardiopulmonary exercise testing*. https://doi.org/10.5281/zenodo.5816170

Quittmann, O. J., Foitschik, T., Vafa, R., Freitag, F., Spearmann, N., Nolte, S., & Abel, T. (under review). *Augmenting the metabolic profile in endurance running by maximal lactate accumulation rate*.

Quittmann, O. J., Schwarz, Y. M., Nolte, S., Fuchs, M., Gehlert, G., Slowig, Y., Schiffer, A., Foitschik, T., & Abel, T. (unpubl.). *Relationship between physiological parameters and time trial performance over 1, 2 and 3 km in trained runners*.