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

Simon Nolte

Part I: Review

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(readxl)
library(rentrez)
library(XML)
library(purrr)
library(ggplot2)
library(MetBrewer)
```

Data retrieval

Read data

Data will be searched from PubMed and Web of Science using the search terms indicated in the preregistration. Search results will be downloaded as a `csv` files (for PubMed entries) and as separate `xls` files (for Web of Science, which has a download limit of 1000 entries). The `xls` files will be manually merged. I developed the following data workflow on a small searching sample with search terms different to the ones of the present study.

```
# file name for PubMed results
pm_file <- NA

# file name for merged WoS results
wos_file <- NA

# read PubMed data
pm_data_pre <- read.csv(pm_file)
pm_data <- data.frame(
  pmid = pm_data_pre[, 1],
  title = pm_data_pre$Title,
  authors = pm_data_pre$Authors,
  journal = pm_data_pre$Journal.Book,
  year = pm_data_pre$Publication.Year,
  doi = pm_data_pre$DOI,
  abstract = NA
)
# read Web of Science data
wos_data_pre <- readxl::read_xls(wos_file)
wos_data <- data.frame(
  pmid = wos_data_pre$`Pubmed Id`,
  title = wos_data_pre$`Article Title`,
  authors = wos_data_pre$`Author Full Names`,
  journal = wos_data_pre$`Source Title`,
  year = wos_data_pre$`Publication Year`,
  doi = wos_data_pre$DOI,
  abstract = wos_data_pre$Abstract
)

# count PubMed results
nrow(pm_data)
# count Web of Science results
nrow(wos_data)
```

Removal of Duplicates

Duplicates will be removed based on their DOI. Before, all entries without a DOI assigned will be excluded. This may remove literature fitting the scope of the review, but as later a random sample of the results is taken, we deem this as acceptable.

```
# Merge data frame
merge_data <- rbind(wos_data, pm_data)

# Filter entries without doi
no_doi <- is.na(merge_data$doi)
merge_data <- merge_data[!no_doi, ]

# Count removal from missing doi
sum(no_doi)

# Find duplicated DOIs
dupl <- duplicated(merge_data$doi)
# Filter duplicates
merge_data <- merge_data[!dupl, ]

# Count removal of duplicates
sum(dupl)
```

Automated title filtering

The titles of all non-duplicated articles will be screened to exclude those that indicate, that an article is not presenting original research (e.g. those with the term ‘review’ in the title). As with the DOI exclusion criteria, this will likely also exclude a minor number of possibly fitting articles, which is again not considered relevant, as later a sample is drawn.

```
# Filter titles by terms
terms <- paste0("review|comment|correction|retraction|meta-analysis",
```

```r
                "|editorial|erratum|reply")
excl <- grepl(terms, merge_data$title, ignore.case = TRUE)
filtered_data <- merge_data[!excl, ]

# Count removal by title filtering
sum(excl)
```

Random Sampling

Based on the included search results, a random sample of 500 articles will be drawn. According to the preregistration, the sample may be later increased in steps of 100 articles if a set threshold of finally included articles is not meet by the initial sample (see `preregistration.pdf` for more details). If the sample is later extended, this will happen by using the same random seed.

For all sampled articles, abstract data will be retrieved. For search results from Web of Science, this data is already included in the search results. For PubMed searches this is not the case, so the abstracts are automatically retrieved by using the article PMID on the PubMed API.

```r
# Assign ids to articles left
filtered_data$id <- seq_len(nrow(filtered_data))

# randomly sample 500 articles
set.seed(4711)
smpl <- sample(filtered_data$id, size = 50)

sampled_data <- filtered_data[smpl, ]
# assign sample id
sampled_data$sid <- seq_len(nrow(sampled_data))

# get abstract from PubMed where missing

no_abstract <- is.na(sampled_data$abstract)

no_abstract_pmid <- sampled_data$pmid[no_abstract]

# function to scrape abstract from the PubMed API
get_abstract <- function(pmid) {
  xml_data <- rentrez::entrez_fetch(
    db = "pubmed",
    id = pmid,
    rettype = "xml",
    parse = TRUE
  )
  text <- XML::xmlValue(XML::getNodeSet(xml_data, "//AbstractText"))
  out <- paste(text, collapse = "///")
  if (out == "") out <- NA
  out
}

# retrieve missing abstracts
sampled_data$abstract[no_abstract] <- purrr::map_chr(
  .x = no_abstract_pmid,
  .f = get_abstract
)
```

Screening

Prepare Title-Abstract plots for screening

To simplify blinded title and abstract screening, I will create a plot for every sampled article, that displays title, abstract and sampling ids. The texts in the plots will have different (randomly assigned) colours, to increase the screener’s focus during the work. If the abstract plot remains empty (i.e. no abstract data could be automatically retrieved), the abstract will be searched online, so in this case the screener will be unblinded.

```r
# Function for creating plots with blinded abstract data
create_abstract_plot <- function(row, data) {
  # get random colour
  # different colours in plots are used to stay concentrated when scanning
  # the abstracts
  clrs <- c(MetBrewer::met.brewer("Moreau")[-4])
  clr <- sample(clrs, 1)

  g <- ggplot(data = NULL, aes(x = 0, y = 1)) +
  ggtext::geom_textbox(
    width = 1,
    box.colour = "transparent",
    label =  gsub(">", "", data$abstract[row]),
    colour = clr
  ) +
  labs(
    title = paste0(strwrap(data$title[row], 80), collapse = "\n"),
    subtitle = paste0("Sampling ID: ", sprintf("%03i", data$sid[row])),
    caption = paste0("ID: ", a$id, "; PMID = ", data$pmid[row])
    ) +
  theme_void()

  ggsave(
    filename = paste0("abstracts/", sprintf("%03i", data$sid[row]), ".png"),
    plot = g,
    dpi = 300, width = 7, height = 5, bg = "white"
  )
}


# Create blinded abstract plots
purrr::walk(
  .x = seq_len(nrow(sampled_data)),
  .f = create_abstract_plot,
  data = sampled_data
)
```

Abstract Screening

Abstract screening results are collected in a `screening_extraction.csv` file. Abstract screening results are inserted in an ‘abstract_exclusion’ column. Based on the exclusion criteria the screener has to answer the following question using the title and abstract data:

1. Does the abstract indicate, that the full-text is not in English language? (if yes, indicate ‘l’; if no, continue)
2. Is the article no original research? (if yes, indicate ‘r’; if no, continue)
3. Is the article not presenting research in humans? (if yes, indicate ‘h’; if no, continue)

4. Does the abstract indicate, that VO2 was not measured (e.g only estimated, approximated, predicted)? (if yes, indicate 'm'; if no indicate 'NA')

```r
# read screening results
incl_data <- read.csv("./screening_extraction.csv")
# assign exclusion criteria as factor levels
incl_data$abstract_exclusion <- factor(
  incl_data$abstract_exclusion,
  levels = c("l", "r", "h", "m"),
  labels = c("no english full-text", "no original research",
             "no humans", "VO2 not measured")
)
# count exclusions
summary(incl_data$abstract_exclusion)
```

Full-text screening

For all articles not excluded during title/abstract scanning the full texts will be downloaded. Results of the full-text screening are inserted in the 'text_exclusion' column of the `screening_extraction.csv` file. Based on the exclusion criteria the screener has to answer the following question using the articles full text:

5. Was no full-text available for the corresponding article? (if yes, indicate 't,' if no continue)
6. Apply step 1-4 from abstract screening (if all questions are answered with yes, continue)
7. Was $\dot{V}O_2$ not continuously measured (e.g. only in the last seconds of the exercise; only post-exercise)? (if yes, indicate 'c'; if no, continue)
8. Did the testing protocol not aspire a subjective exhaustion of the participants (e.g. involved only sub-maximal exercises)? (if yes, indicate 'e'; if no, continue)
9. Did the testing protocol have a mean duration of more than 20 minutes until exhaustion (e.g. long graded incremental exercise protocols, long time-trials)? (if yes, indicate 'd'; if no, continue)
10. Are there other rationale to exclude the article (if yes, indicate 'o' and state the exact reason in the 'exclusion_note' column; if no indicate 'NA')

```r
# assign exclusion criteria as factor levels
incl_data$text_exclusion <- factor(
  incl_data$text_exclusion,
  levels = c("t", "l", "r", "h", "m", "c", "e", "d", "o"),
  labels = c("no fulltext", "no english full-text", "no original research",
             "no humans", "VO2 not measured", "VO2 not continously measured",
             "no exhaustion", "long protocol", "other")
)
# count exclusions
summary(incl_data$text_exclusion[is.na(incl_data$abstract_exclusion)])
```

Data Extraction

Data will be extracted from all articles not excluded in the previous steps. If one of the question cannot be answered based on the full-text, this will be indicated by 'NA' in the corresponding column. If an article references to other studies or supplementary material for further information on the methods, these additional materials will be assessed for data extraction.

The screener has to answer the following question during the data extraction process and has to fill the answer into the corresponding column of the `screening_extraction.csv` file:

'cart': Which metabolic cart did the authors use? State manufacturer and model.

‘type’: Which type of data was processed? Either ‘bbb’ for breath-by-breath data, ‘mc’ for mixed chamber, or others.

‘pre’: Which preprocessing algorithm did the researchers use to calculate the breath-by-breath data?

‘software’: Which software did the researchers use to analyse the gas exchange data? State the name and the version.

‘interpolation’: Which interpolation procedure did the researchers use to process the raw data? E.g. ‘second’ for interpolation to full seconds.

‘ptype’: Which type of data processing did the researcher use when analysing the gas exchange data? Either ‘ta’ for time-based average, ‘ba’ for breath-based average, ‘bw’ for Butterworth filter, or other

‘averaging’: Which type of averaging did the researchers use, if they used an averaging strategy for data processing? Either ‘m’ for moving, ‘b’ for binned, or others.

‘parameters’: Which parameters did the researchers use for the data processing. E.g. the length of the interval for calculating a breath- or time-based average, or the parameters (order, cut-off frequency) for a digital filter.