Data Wrangling Workshop

McGill initiative in Computational Medicine

Sean Nesdoly

2021-04-15

Contents

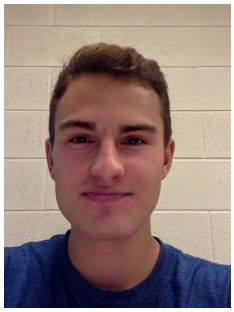
1 Introduction

This workshop will introduce the fundamental principles behind data wrangling as well as present some of the best practices for implementing these concepts in both the R and Julia programming languages. Specifically, it will cover: (1) exploration of raw data from the genomics domain; (2) identification of applicable data structures specific to the task at hand; (3) 'wrangling' or 'munging' of data into the selected data structure using the R `tidyverse` packages; (4) cleaning the structured data, as needed; and, (5) transforming the 'wrangled' data into a format suitable for downstream analyses or archival storage, with reusability in mind. An alternative formulation of basic data wrangling techniques will also be presented in Julia, demonstrating that these concepts are generalizable between programming languages.

1.1 Main Objectives

1. To understand the fundamental principles behind data wrangling, cleaning, & munging.
2. To become proficient with data manipulation in R using the `tidyverse` packages.
3. Familiarize yourself with Julia, a general purpose, high performance, open source programming language for computational medicine and beyond.

1.2 Instructor Biography



I'm Sean, a passionate programmer with a love for biology. I recently graduated from the Biological & Biomedical Engineering program at McGill University with an M.Eng. degree that focused on bioinformatic algorithm development; my undergraduate training was in Computer Science. I am originally from Calgary, Alberta. Outside of work, I enjoy hiking, cycling, hockey, and cooking new dishes (attempting to, at least). I am always excited to learn new things, and to teach!

2 Prerequisites

Complete the following before attending the workshop:

1. Install R (version 3.5+): https://utstat.toronto.edu/cran/
2. Install RStudio Desktop (the free version): https://rstudio.com/products/rstudio/download/
3. Install all R packages contained within the `tidyverse`; to do so, execute the following lines in R:

```
install.packages("tidyverse")
library(tidyverse)
```

4. Install Julia: https://julialang.org/downloads/
5. Clone or download the following git repository from GitHub: https://github.com/SeanNesdoly/MiCM-Data-Wrangling-Workshop
   - To **clone** the repository, execute the below code from a terminal. When git prompts for your password, enter your GitHub personal access token (see here for details).

```
# Replace the below file path with your own
cd ~/path/to/working/directory/
git clone https://github.com/SeanNesdoly/MiCM-Data-Wrangling-Workshop.git
```

- To **download** the repository, look for the green button labelled 'Code' and click on 'Download ZIP'. Once it downloads, extract the ZIP file to your desired location.

2.1 TODO Alternative: Docker?

3 TODO Introduction to Data Wrangling

Also known as 'munging'.

4 TODO Exploring Raw Data

5 TODO Data Wrangling with the R `tidyverse` Packages

- Compute Canada Advanced Training @ Queen's (~/qbmcoh/cac/)
- /Users/sean/qbmcoh/duanlab/patient-classification/R
- /Users/sean/qbmcoh/duanlab/rproject-template
- /Users/sean/Documents/r-notes

5.1 Data Manipulation with `dplyr`

```
mutate()
select()
filter()
summarise()
arrange()
```

5.2 Piping with `magrittr`

```
f(x)
x %>% f
```

5.3 The `tibble` Data Frame

- Date Frames: 'They do less and complain more'
- Enhanced print method for large datasets
- Keeps data in its raw format

```
tibble(x = 1:5, y = 2, z = x^2 + y)
```

5.4 Functional Programming with `purrr`

```
map()
```

6 TODO Cleaning Structured Data

7 TODO Choosing a Data Structure

8 TODO Preparing Data for Downstream Analyses

9 TODO Introduction to Julia

- The ability to add explicit type annotations into code improves human readability and catches errors upfront, both of which are issues in R.
  - For example, 'function foo(num::Int)::String' defines a function 'foo' that takes an integer 'num' and returns a string.

10 TODO Practical Assignment

Within the genomics/omics domain. Ideas:

- Genome in a Bottle (https://www.nist.gov/programs-projects/genome-bottle)
- The Cancer Genome Atlas (https://portal.gdc.cancer.gov/)
- Case vs. control datasets, with patient phenotypes; often very messy, so would be a good candidate for 'wrangling' & cleaning.
- Microbiome analysis (contains taxonomy assignments, sample metadata, etc.)

11 Resources

11.1 General

- https://en.wikipedia.org/wiki/Data_wrangling
- OOP vs. multiple dispatch: methods are owned by functions and not by objects

11.2 R

- https://r4ds.had.co.nz/tidy-data.html
- https://en.wikipedia.org/wiki/R_(programming_language)
- https://r-pkgs.org/
- http://adv-r.had.co.nz/Introduction.html
- https://github.com/GreenwoodLab/knitr-tutorial
- https://swcarpentry.github.io/r-novice-inflammation/15-supp-loops-in-depth/

11.3 Docker

- https://www.rocker-project.org/
- https://environments.rstudio.com/docker
- For code: `RUN git clone gitURLorSSH`

11.4 Polling Software

- TurningPoint: https://www.mcgill.ca/polling/

11.5 Common Errors

11.5.1 XQuartz on macOS

Some R packages require XQuartz to be installed. If you run into this error, download & install a stable version of XQuartz and restart your computer. Alternatively, if you use Homebrew as your package manager, you can run the following from a terminal:

```
brew install --cask xquartz
```