IntroClass: Stat4600-MBIO7160

Aleeza Gerstein

2019-09-11

Hopefully this is the only class I will have to do this

Course objectives:

1. To understand central concepts in modern biostatistics
2. To utilize the R Programming language to apply these statistics to biological data
3. To build a research toolkit:
    - apply techniques for reproducible research
    - develop discussion participation techniques
    - practice science communication skills

Course evaluation:

40% Preparation for an participation in discussions 20% Discussion moderation 20% Final project 10% Final presentation 10% Peer evaluation

The primary goal of this class is to *learn someone useful*. As a group we should decide on the course logistics to best meet that goal.

Logistics to discuss today

1. Students auditing/dropping-in
2. Class timing
3. Schedule of student facilitation
4. Getting started with Git
5. Reproducible research with Git, R studio projects, the `here()` package, and R markdown

Students auding/dropping-in

A number of graduate students have expressed interest in auditing or dropping-in on these classes. My initial instinct is typically 'the more the merrier' but that might not be the correct way forward in this case. Since it will be students who will be primarily leading the lectures, I think it is up to you to decide whether it will be potentially disruptive to have people sitting in that do not have as vested an interest in doing the work outside of class that you will all be expected to do. We could also set-up parameters (e.g., must attend all classes) for their involvement. I think this should be a unanimous decision and ask you to be candid about your thoughts: if even a single person wants to keep the group to enrolled students only then this is what we will do.

Class timing

There is a conflict with a number of Bannatyne students, who will have another obligation until 14:20. Starting in October we will move class to 13:30. The original intent was to meet every two weeks, which I already thought would be tight. An option is to meet more regularily (2 of every 3 weeks?), and potentially having some classes at Bannatyne. Let's discuss (following the information provided below).

Schedule of student facilitation

(depends on what was decided above). The goal is to work through at least six chapters. I listed Chapters 1-5 and 8 on the syllabus but this can be adjusted. *Generative models for discrete data* (number of mutations, epitope detection) *Statistical modeling* (DNA base pair counts, Hardy-Weinberg equlibrium, haplotype frequencies) *Mixture models* (ChiP-Seq data, next generation sequencing read counts) *Clustering* (single cell RNA-seq, flow cytometry, cell clustering, 16S metagenomics) *Multivariate analysis* (species abundance, mass spectroscopy) *High-throughput count data* (RNA-seq)

If we do six chapters and meet every two weeks that takes us to the last day of the semseter (December 4).

Facts to know: there are now seven students enrolled and I think we should also have a dedicated class for student presentations.

Back to timing discussion (which can be an ongoing discussion/revisited later in the semester)

Getting started with Git

We are going to use GitHub for our course notes and assignment

Git/Github is a way to manage version control, in the cloud. In the words of Jenny Bryan "Git manages the evolution of a set of files – called a repository – in a sane, highly structured way. If you have no idea what I'm talking about, think of it as the "Track Changes" features from Microsoft Word on steroids." GitHub is the webhosting service we will use for Git (others do exist e.g., Bitbucket, GitLab, but GitHub is the most common).

For our class we're going to have everyone set up a dedicated repository, and ask that all the work you do for the class is kept there (and kept current).

To do this we are going to follow Jenny's amazing "Happy Git and GitHub for the useR" book. Follow the "pre-workshop" setup: https://happygitwithr.com/workshops.html

After the last step ("Connect RStudio to Git and GitHub") you should follow Chapter 15 "New project, GitHub first"). This will serve as your 'landing pad' for the class: you will now have a github repo and a dedicated R project for the course. This is mine: https://github.com/acgerstein/ModernStatsModernBiol

Reproducible research with Git, R studio projects, the `here()` package, and R markdown

R projects

We're now going to switch to a second amazing resource that Jenny Bryan developed as a two-day workshop, "What They Forgot to Teach You About R": https://whattheyforgot.org/

From Chapter 2: Chapter 2 Project-oriented workflow

This let's us move away from using `setwd()` which is entirely unreproducible. e.g.,

```
setwd("/Users/acgerstein/Nextcloud/Umanitoba/Teaching/2019MBIO7160-ModernStats/ModernStatsModernBiol/")
mbpoli <- read_csv("data/2019MBpoli.csv")
```

```
## Parsed with column specification:
## cols(
##   Riding = col_character(),
##   Majority = col_character(),
##   Party = col_character(),
##   IncumbentElected = col_double(),
##   NewRiding = col_double(),
##   Winnipeg = col_character(),
##   Gender_elected = col_character(),
##   Gender_incumbent = col_character(),
##   PC = col_double(),
##   NDP = col_double(),
##   Lib = col_double(),
##   Green = col_double(),
##   Ind = col_double()
## )
```

The solution is to use R Studio projects (.Rproj).

Jenny Bryan: https://whattheyforgot.org/project-oriented-workflow.html

Don't nest .proj files! Because it interferes with `here()` …

The here package

Unsuprisingly (at this point) see posts by Jenny Bryan (https://github.com/jennybc/here_here, https://whattheyforgot.org/safe-paths.html) and Malcolm Barrett (https://malco.io/2018/11/05/why-should-i-use-the-here-package/)

RStudio projects let us set up a local working directory, which makes it easier for someone else to access your files with the same folder and file structure.

The here package let's us write file paths that work across operating systems: it detects the root directory and writes let's us build paths accordingly. If you are familiar with `file.path()` it is similar, except more robust ouside of projects.

Importantly the root directory is *not* where your `*.rmd` script is! It is where your `*.Rproj` file is! Combined with git, this means that everything can be entirely reproductible.

```
getwd()
```

```
## [1] "/Users/acgerstein/Nextcloud/Umanitoba/Teaching/2019MBIO7160-ModernStats/ModernStatsModernBiol/cl
```

```
dir()
```

```
## [1] "190911IntroClass.log" "190911IntroClass.pdf" "190911IntroClass.Rmd"
```

```
#load the here package
here("data", "2019MBpoli")
```

```
## [1] "/Users/acgerstein/Nextcloud/Umanitoba/Teaching/2019MBIO7160-ModernStats/ModernStatsModernBiol/da
```

```
MBpoli <- read_csv(here("data", "2019MBpoli.csv"), col_type = cols())
MBpoli
```

```
## # A tibble: 57 x 13
##    Riding Majority Party IncumbentElected NewRiding Winnipeg Gender_elected
##    <chr>  <chr>    <chr>            <dbl>     <dbl> <chr>    <chr>
##  1 Aggas~ yes      PC                   1         0 other    F
##  2 Assin~ no       PC                   1         0 Winnipeg M
##  3 Borde~ yes      PC                   0         1 other    M
##  4 Brand~ yes      PC                   1         0 other    M
##  5 Brand~ yes      PC                   1         0 other    M
##  6 Burro~ no       NDP                  0         1 Winnipeg M
##  7 Conco~ yes      NDP                  1         0 Winnipeg M
##  8 Dauph~ yes      PC                   1         0 other    M
##  9 Dawso~ yes      PC                   1         0 other    M
## 10 Elmwo~ no       NDP                  1         0 Winnipeg M
## # ... with 47 more rows, and 6 more variables: Gender_incumbent <chr>,
## #   PC <dbl>, NDP <dbl>, Lib <dbl>, Green <dbl>, Ind <dbl>
```

Bonus Jenny Bryan approved hill I will die on: How to name files

https://speakerdeck.com/jennybc/how-to-name-files

R markdown

It's great. Use it. https://rmarkdown.rstudio.com/lesson-1.html