MCB 536A: TFCB
Introduction to Tools for Computational Biology

Today’s objectives

After today’s class, you should be able to:

- Locate information relevant to the course (lecture materials, assessment, communication streams)
- Identify range of skills and concepts covered in this course
- Install software required for this course

Zoom logistics

- Same Zoom link for duration of class; room is always open for additional calls
- Basic etiquette:
  - stay muted; type `/hand` in the chat window if you'd like to speak
  - type questions/concerns into chat window (these don't persist after the call ends); send to everyone by default or message instructor privately if necessary

Introductions: Kate

- Leader of Training and Community for data-intensive research at Fred Hutch
- Research background in genomics, bioinformatics, evolutionary biology (plants, Drosophila); current research in cancer genomics
- Favored programming tools are unix/bash and R

Introductions: other instructors

Maggie Russell (TA)

Trevor Bedford

Phil Bradley

Jesse Bloom

Will Hannon (TA)

Erick Matsen

Gavin Ha

Arvind Rasi Subramaniam

Introductions: You!

- Name (including preferred form of address)
- Research interests (type of data, model organism, research questions, etc)

Course objectives

By the end of the course, you should be able to:

- Code in R, Python, and Unix/bash shell scripting using appropriate syntax and code convention
- Select appropriate tools to perform specific programming and data analysis tasks
- Apply good practices for computational research, including project organization and documentation
- Analyze common forms of data generated by molecular biology experiments including high throughput sequencing, flow cytometry, and 96-well plate readers.

Course materials:
Syllabus, lectures and demos

Rendered materials (prettier/easier to view):
https://fredhutchio.github.io/tfcb_2020/

Original GitHub repository:
https://github.com/fredhutchio/tfcb_2020

Course materials: Assignments and grading

Canvas:

MCB 536 A Au 20:

Tools For Computational Biology

Eight assignments (10% each)

+ participation (20%)

Course materials: Communication

TFCB 2020 (MCB 536A)

- #general: course announcements (please turn on notifications for this channel!)
- #lectures-homework: questions about course content and help for homework
- see pinned posts in each channel for quick links and reminders!

Finding help

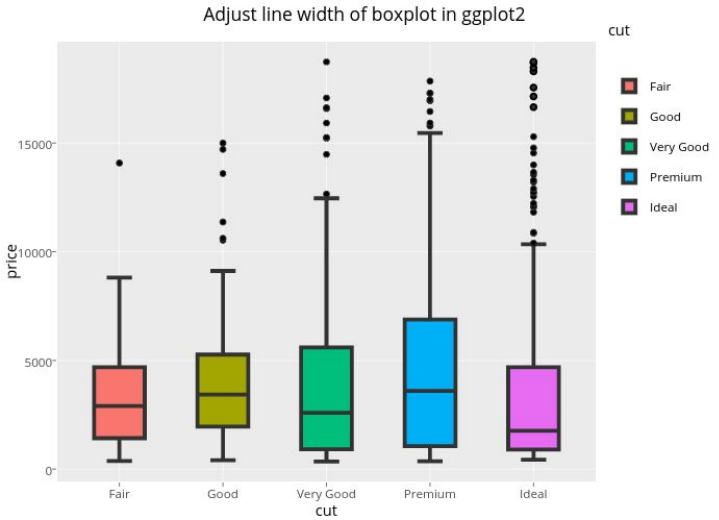
- We are working with open source tools this semester, so there are lots of resources to help you, freely available online, though it does take practice to help yourself effectively!
- Homework you submit should be in your own words, with a citation (inline comment) of the online source or person that helped you

Questions you will be able to answer after this course:

- What are the most common tools in computational biology?
- How are biological data (from molecular biology research) represented?
- What are appropriate methods for making computational work reproducible?

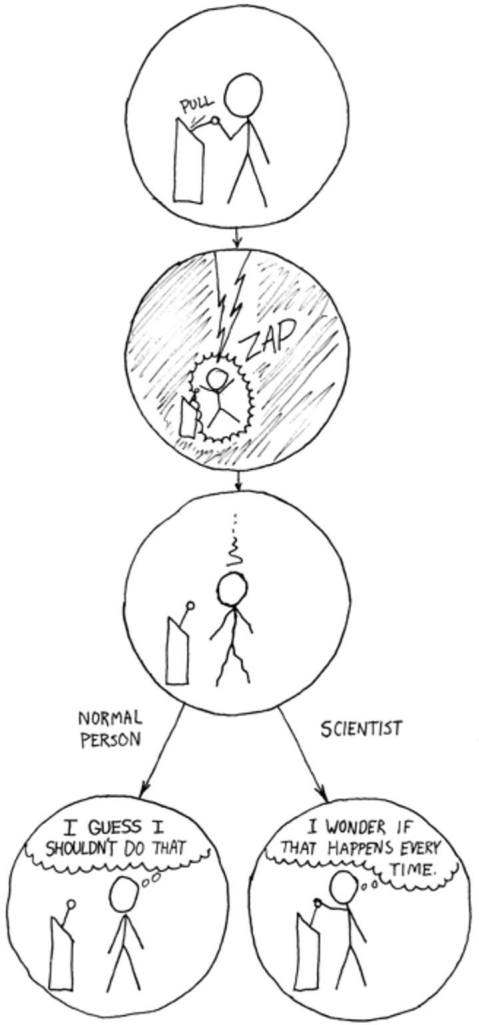
Reproducible Computational Biology


```
p Soccer Data 'soccer.txt" ######

afile into R ###########
able("soccer2002.txt", header=TRUE)

ze ####
occer$Freq)

an ####
leN)*(sum(soccer$Go

#### Poisson probability for X=1,
dpois(1, lambda=smean)

#### Poisson probabilities for X=
prob<-dpois(0:8, lambda=smean)
prob


#### Compute expected frequencies
efreq<-sampleN*prob
efreq
```



Reproducible computational methods

- *Reproducibility*: obtaining the same results multiple times
  - Confirm previously published scientific results
  - Automate large-scale data analysis projects
- *Transferability*: using methods multiple times
  - Among researchers
  - Among research questions



Reproducible computational biology: Technical skills

Unix command line/remote computing

Git and GitHub

Python

R statistical programming

Reproducible computational biology: Concepts

Good Enough Practices in Scientific Computing

1. **Data management**: saving both raw and intermediate forms, documenting all steps, creating tidy data amenable to analysis.
2. **Software**: writing, organizing, and sharing scripts and programs used in an analysis.
3. **Collaboration**: making it easy for existing and new collaborators to understand and contribute to a project.
4. **Project organization**: organizing the digital artifacts of a project to ease discovery and understanding.
5. **Tracking changes**: recording how various components of your project change over time.
6. **Manuscripts**: writing manuscripts in a way that leaves an audit trail and minimizes manual merging of conflicts.

At the end of this class:

- You will have building blocks upon which to start constructing your own reproducible computational research
  - Technical skills: software usage and programming languages
  - Concepts: an understanding of how to apply these skills effectively
- You will apply these skills and concepts in a capstone project (writing code to analyze a previously published dataset)

After this course, you may NOT be able to:

Use ALL of the tools your research will require (especially those that are GUIs)

Know the best algorithm or analysis method for a specific research question

Code with expert-level skills

...but you should be equipped to achieve these goals later.

Summary

- This course focuses on data-driven computation using open-source scientific tools
- Course materials will reflect use of these tools
- Regardless of the extent to which you use computation in your research, learning the methods will make you a better scientist

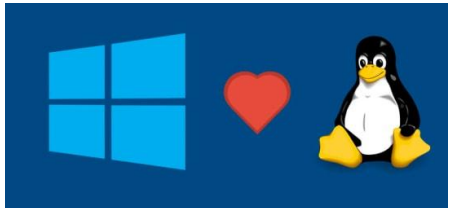
For next time:

- Compete the pre-class survey
- Install all required software (and get a GitHub account!)
- Orient yourself to recommended reading materials
- Ensure your paperwork has been submitted to the Hutch so you can obtain a HutchNetID!

Next time: Unix command line!

Please complete the pre-class survey
(it should take less than 5 minutes)

If you have any questions, also feel free to add those to the chat.
When you're done with the survey, start checking to see necessary software is installed.



Required software
(order of usage during course):

Windows Subsystem for Linux (not Mac)

GitHub account (share username in survey),
GitHub Desktop App (plus command line tools)

Text editor, spreadsheet program

Anaconda, plus extra libraries

R and RStudio, plus extra packages

Installation instructions here