# FOUNDATIONS OF TRANSLATIONAL BIOMEDICAL INFORMATICS

USC University of Southern California

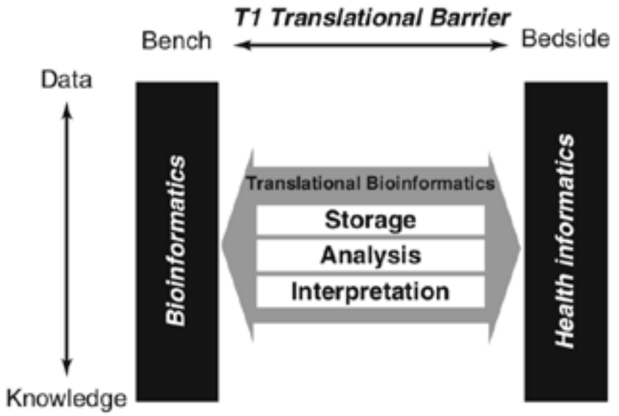
USC Institute Of Translational Genomics
Keck Medicine of USC

# TRANSLATIONAL BIOINFORMATICS / BIOMEDICAL INFORMATICS

- Translation of biological ("bench") discoveries into actual impact on clinical care ("bedside") and ultimately on population health

## Definitions

- The American Medical Informatics Association (AMIA) defines Translational Bioinformatics as "the development of storage, analytic, and interpretive methods to optimize the transformation of increasingly voluminous biomedical data, and genomic data, into proactive, predictive, preventive, and participatory health."



Jessica D. Tenenbaum, Nigam H. Shah, and Russ B. Altman

# DATA SCIENCE

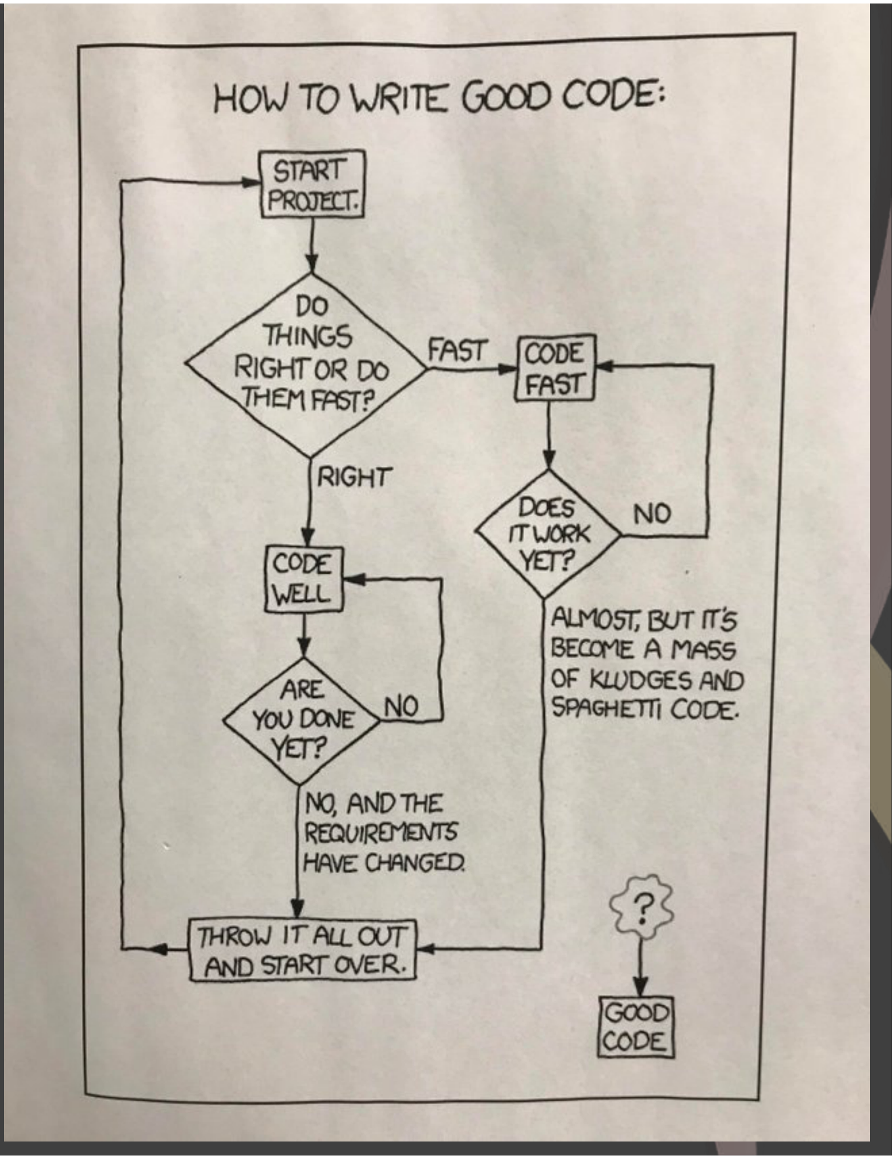
## Data in the context of bioinformatics?

- Biomedical informatics involves big data. One of the biggest drivers lately is genomics and sequencing. Lets go with that example. Aligning a genome is a problem that is easy to break up. Aligning 1 read doesn't impact aligning another read. If I have 1 billion reads, and a single CPU can align 2500 reads per second it will take 72 hours. If you want the answer quicker you need to split the analysis up and send it off to different computers that ideally all share network storage so that we can keep track of data. This is solved by High-Performance-Computing.

## At this point, we should highlight there are three major areas of HPC that bioinformatics sees:

- (1) Problems that can be easily broken up – divide and conquer. Analysis of genomes is a great example. Divide and conquer works well when the core calculation doesn't require knowing what the result is from other calculations. For example, aligning 1 read to a genome, frequently (except for assembly), aligns to the same position in the genome regardless of what everyother read aligns.
- (2) Problems that require lots of communication at each step & are less easy to break up.

# Knowing The Objective; Biomedical Informatics Is Often Driven By 1 Offs


HOW TO WRITE GOOD CODE:
START PROJECT.
DO THINGS RIGHT OR DO THEM FAST?
FAST
CODE FAST
RIGHT
DOES IT WORK YET?
NO
CODE WELL
ARE YOU DONE YET?
NO
ALMOST, BUT IT'S BECOME A MASS OF KLUDGES AND SPAGHETTI CODE.
NO, AND THE REQUIREMENTS HAVE CHANGED.
THROW IT ALL OUT AND START OVER.
?
GOOD CODE

# What to Expect

# Course Overview

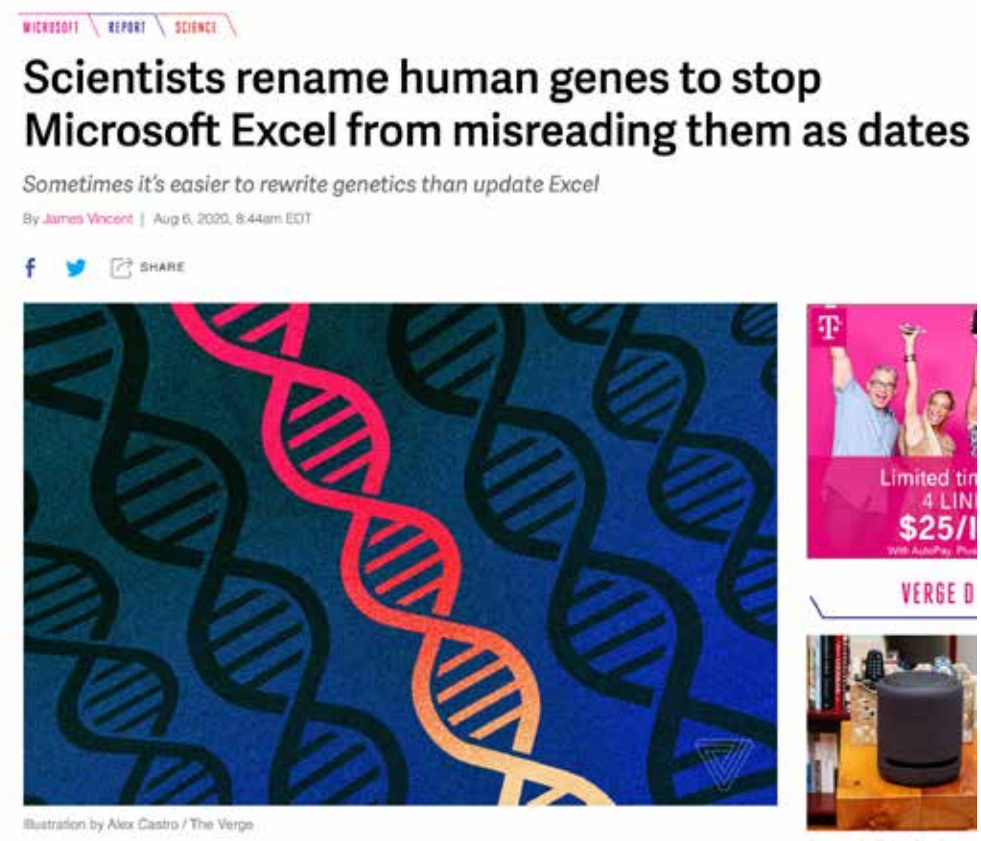
- The objective of this course is to train individuals with strong backgrounds in biological or medical fields the analytical and computational skills for analysis of biomedical data.
- It will introduce students to tools and concepts that will be instrumental throughout the program. Particular focus will be on applicability to the healthcare field and training students to effectively implement, develop, and design bioinformatic solutions within different healthcare applications from prototyping to production. They will be trained and have an understanding of modern molecular data with a major emphasis on data analysis and data processing associated with next-generation sequencing data.
- This course targets individuals who have laboratory experience generating biomedical data, and aims to provide them with the foundations, basic principles, and core concepts in scripting and computing that are necessary in biomedical informatics. The course will focus on teaching by example with the understanding that applied biomedical informatics frequently favors rapid and iteratively developed single-use analyses that must be both reproducible and documentable, for example how to work within a command-line based environment, basic scripting such as with R, bash, or javascript. They will learn versioning, unit testing, and prototyping focusing on being able to rapidly analyze and explore datasets. They will learn the fundamentals of web-applications, biomedical databases, and on-line resources and how to utilize and integrate these within one's own analyses using APIs and connectors. It will also include an overview of regex and web mining, data-types and data structures, program flow, versioning and best practices. R will be utilized throughout as part of the course to familiarize students with various programming tools. High performance computing will be introduced from a user perspective along with best practices.
- This course is an introductory level course and restricted for Masters of Science (MS) degree in Translational Biomedical Informatics. This course is not intended for those experienced in command-line tools, scripting, database, and web-based applications. This course and the timing of core concepts will complement companion courses provided at the same time.

# Learning Objectives

- The goal of this introductory platform course is to teach core fundamentals that will allow someone trained in biology or medicine to use modern computing and bioinformatics tools to rapidly and reproducibly answer biological questions within an applied setting. The focus is not on teaching how to developing tools, modules, or frameworks for community distribution, and more focused on how researchers can use existing tools together to explore novel biomedical questions in ways that retain reproducibility (such as through versioning). Still students will learn how to interface and communicate with development teams, and gain a basic understanding of project management frameworks.
- While the course teaches using the statistical framework R, it does not teach fundamental of statistics and presumes students have had an undergraduate or graduate level biostatistics or biometrics course. Application of statistical approaches will be taught on how they can be deployed using defined software and data, though this course does not teach statistical interpretation or design of analyses as that is beyond the scope. Several electives are available should students wish to supplement and gain further expertise.
- Upon successful completion of this course, students will be able to: interface with the command line; utilize versioning tools following best practices; create basic scripts in R, create basic web-apps in R-Shiny; describe the basic concepts of data-types and data structures and when to use them; effectively use best practice high performance computing concepts.

# REQUIREMENTS

- This course has specific hardware and software requirements as part of the Master’s in Translational Biomedical Informatics Program. In order to optimize the ability for students to work together with uniformity there are specific computing hardware requirements. Students will be required to have a 2013+ MacBook, MacBook Air, MacBook Pro, iMac or Mac Pro with version 10.12+ (macOS) with a minimum of 4 Gbytes of RAM. The course will require a suite of open-source software that will be provide 2 weeks prior to the start of courses. Students will need to have video and audio capabilities that typically come with most MacOS computers.



MICROSOFT \ REPORT \ SCIENCE

## Scientists rename human genes to stop Microsoft Excel from misreading them as dates

*Sometimes it's easier to rewrite genetics than update Excel*

By James Vincent | Aug 6, 2020, 8:44am EDT

Illustration by Alex Castro / The Verge

# Description of Assignments

- Informatics conducted in lecture halls are inherently difficult, especially for those who do not have prior experience. This course forms the framework with other concurrent courses, and early participation will be essential. The work load for this course will complement other concurrent courses, and the work-load expectations will be front-loaded to insure the foundations are provided within the first half of the course.
- While content will be available on-line assignments and coursework requires timely iterative completion. It will be difficult to catch-up, and teamwork necessitate that deadlines cannot be individually altered.
- There will be a month long final project on a topic of student's choice either on their own or within a group aimed at mimicking a bioinformatics analysis, and communication to collaborators. The deliverables are a project scope of work, a written report of the data analysis in an R Markdown, a website, and a two-minute video communicating what the group learned. The project proposal described the motivation for the project, the project objectives, a description of the data, how to obtain the data, an overview of the computational methods proposed to analyze the data and a timeline for completing the project.

# GRADING

- □ 30% Assignments. Assignments are typically weekly/bi-weekly with specified due dates. Assignments late by 1 week or less receive 80%; Assignments late 2 weeks or less receive 50% credit.
- □ 20% Quiz. Bi-Weekly concept exams
- □ 10% Midterm Exam. A midterm exam constitutes 10% of the grading covering key concepts.
- □ 25% Final project. A final project consisting of a web application demonstrating multiple aspects of the course. The final project will be broken down into 1/4th initial proposal, 1/4th initial functional prototype, 1/4th proposal, and 1/4th functional application.
- □ 15% Final Exam. A final exam constitutes 10% of the grading covering key concepts.

# TIME COMMITMENT

## Historically, many people spend 20 hours+ week.

- Most people 20+ hours, said that it was by their choice. A substantial portion have said that this course was one of the most valuable they took. That it was not because of the teacher. It was the content and love of data.

## This is graduate course – there are points you will struggle

- I expect you to try to figure out a problem when you are stumped at least 4 hours. However, never more than 8. The problem is 95% typos.

## The goal to train you how to solve problems

# SCHEDULE

## August 18/20

- Introduction to course objectives/requirements, teaching approaches, and structure
  - Installation of key software and setup of individual work environments.
- *Navigation of remote servers using a Unix-like environment.*
  - Command line basics: Basic commands, navigating servers, connecting to databases, and review. Manipulating, editing, and inspecting files. Introduction to filesystems and permissions.
  - Scripting in BASH, with program control, further exercises in program control.
  - Data types in BASH and program control
- Homework Week 1 Part 1 Due 8/20 11:59AM PST
- Homework Week 1 Part 2 Due 8/23 11:59PM PST

# SCHEDULE

## August 25/27

- **BASH Scripting** continued
- Introduction to HTML,CSS, JS, and webpages, cloud-computing.
- Communicating in advanced environments, tunneling, high-performance computing job scheduling. Bash scripting, variables, data-types.
- Homework Week 2
- Quick Week 2

## Sept 1/3

- Python I
- Homework Week 3

## Sept 8/10

- Python (cont)
- Graphing (Vega/D3.js)
- Homework Week 4
- Quiz Week 4

## Sept. 15/17

- Advanced concepts in dataflow, authentication and webservers
- Databases. Introduction of relation and non-relational databases, joining and integrating datatypes, continued data wrangling.
- Homework Week 5

## Sept. 22/24

- Javascript; introduction to D3.js, plotly, and web-app frameworks.
- Integration within workflows, and data-processing management and flow. Introduction to HPC best practices. Introduction to cloud computing
- Homework Week 6
- Quiz Week 6

## Sept. 29 & Oct 1

- Catchup
- Midterm

## Oct 6/8

- Diving into R for scripting and analysis.
- Plotting and visualization in R.
- Homework Week 7

## Nov 3/5

- Practical Examples
- Homework Week 11

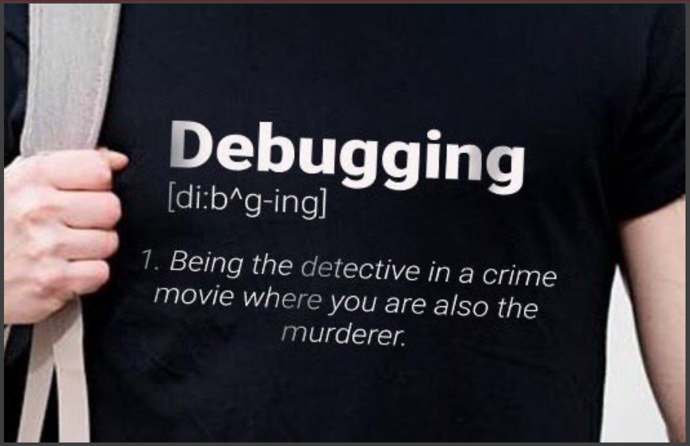
## November 9-13.

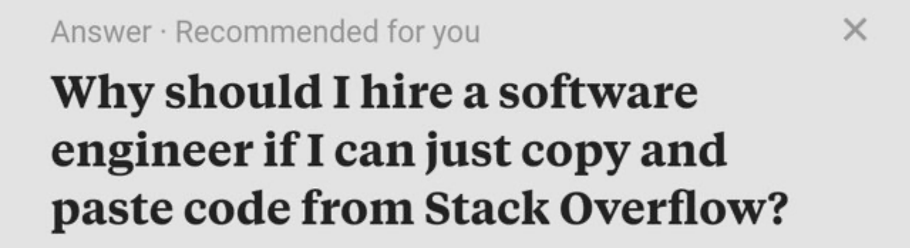
- Final Projects Due.

## Nov 17th week

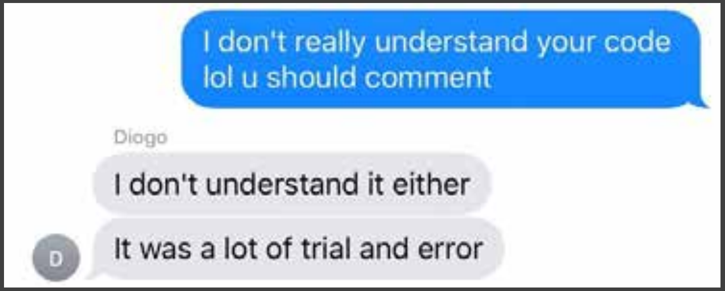
- Final Exam

# Always Start With Something That Works


Debugging
[di:b^g-ing]
1. Being the detective in a crime movie where you are also the murderer.


Answer · Recommended for you

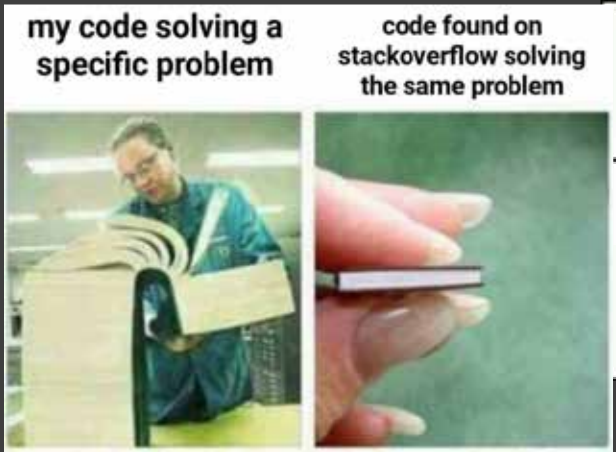
**Why should I hire a software engineer if I can just copy and paste code from Stack Overflow?**

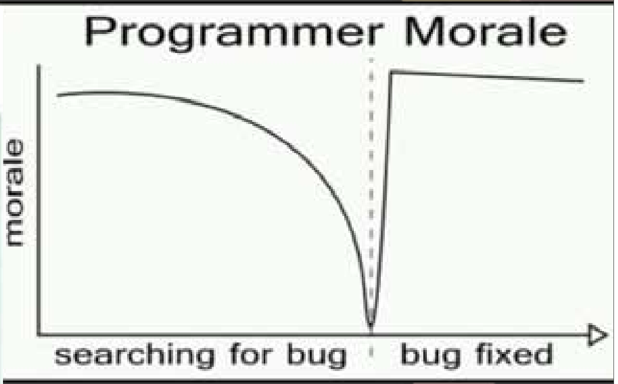

I don't really understand your code lol u should comment

Diogo

I don't understand it either

It was a lot of trial and error

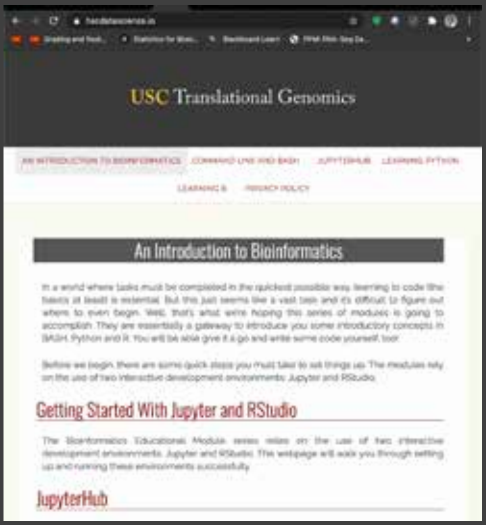

my code solving a specific problem
code found on stackoverflow solving the same problem



# DIVING IN

https://www.hscdatascience.io/

# 5+ Learning Modes



USC Translational Genomics

AN INTRODUCTION TO BIOINFORMATICS · COMMAND LINE AND BASH · JUPYTERHUB · LEARNING PYTHON · LEARNING R · PRIVACY POLICY

## An Introduction to Bioinformatics

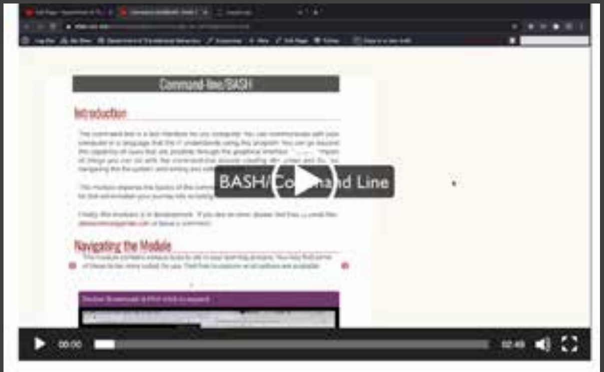
In a world where tasks must be completed in the quickest possible way, learning to code (the basics at least) is essential. But this just seems like a vast task and it's difficult to figure out where to even begin. Well, that's what we're hoping this series of modules is going to accomplish. They are essentially a gateway to introduce you some introductory concepts in BASH, Python and R. You will be able give it a go and write some code yourself, too!

Before we begin, there are some quick steps you must take to set things up. The modules rely on the use of two interactive development environments: Jupyter and RStudio.

### Getting Started With Jupyter and RStudio

The Bioinformatics Educational Module series relies on the use of two interactive development environments: Jupyter and RStudio. This webpage will walk you through setting up and running these environments successfully.

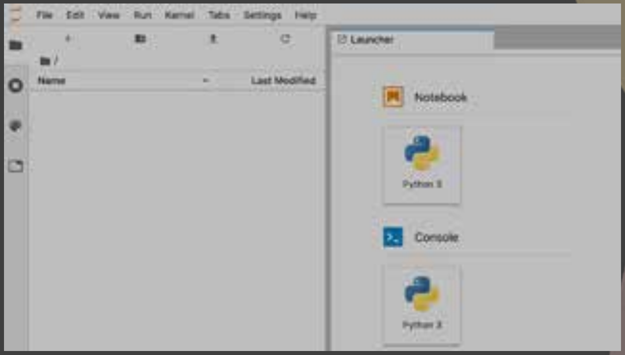
### JupyterHub



## Command-line/BASH

### Introduction

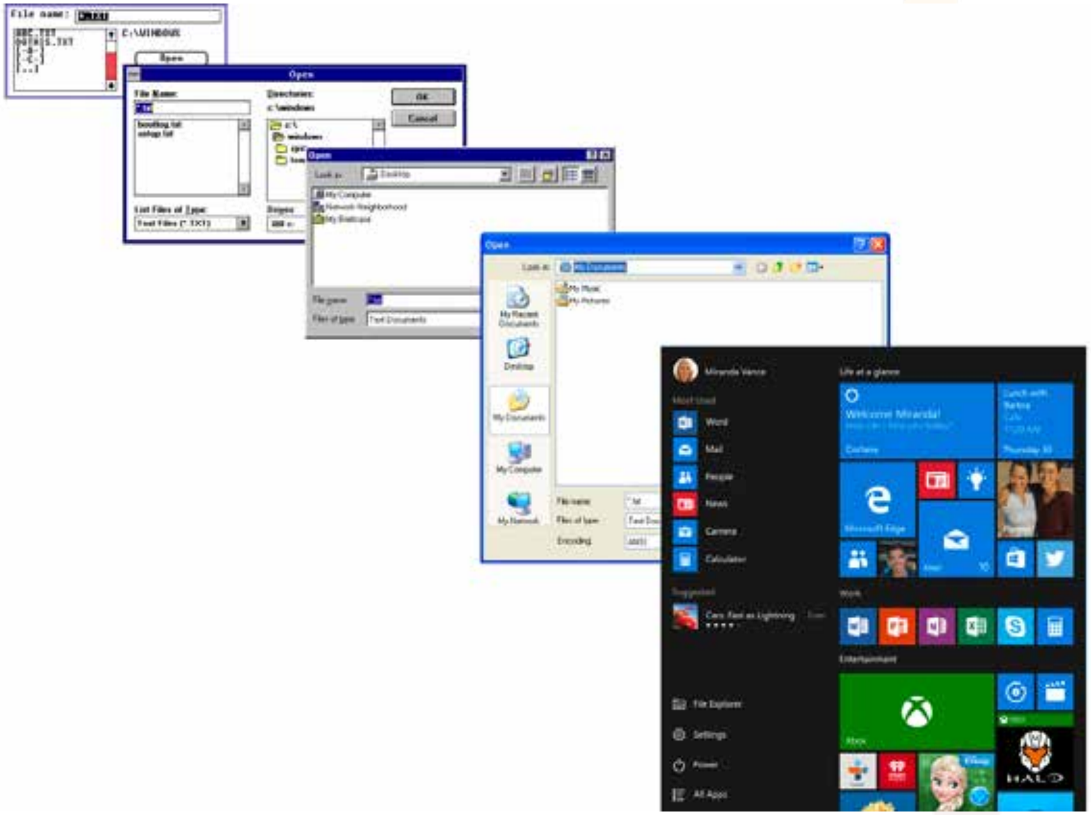
### Navigating the Module



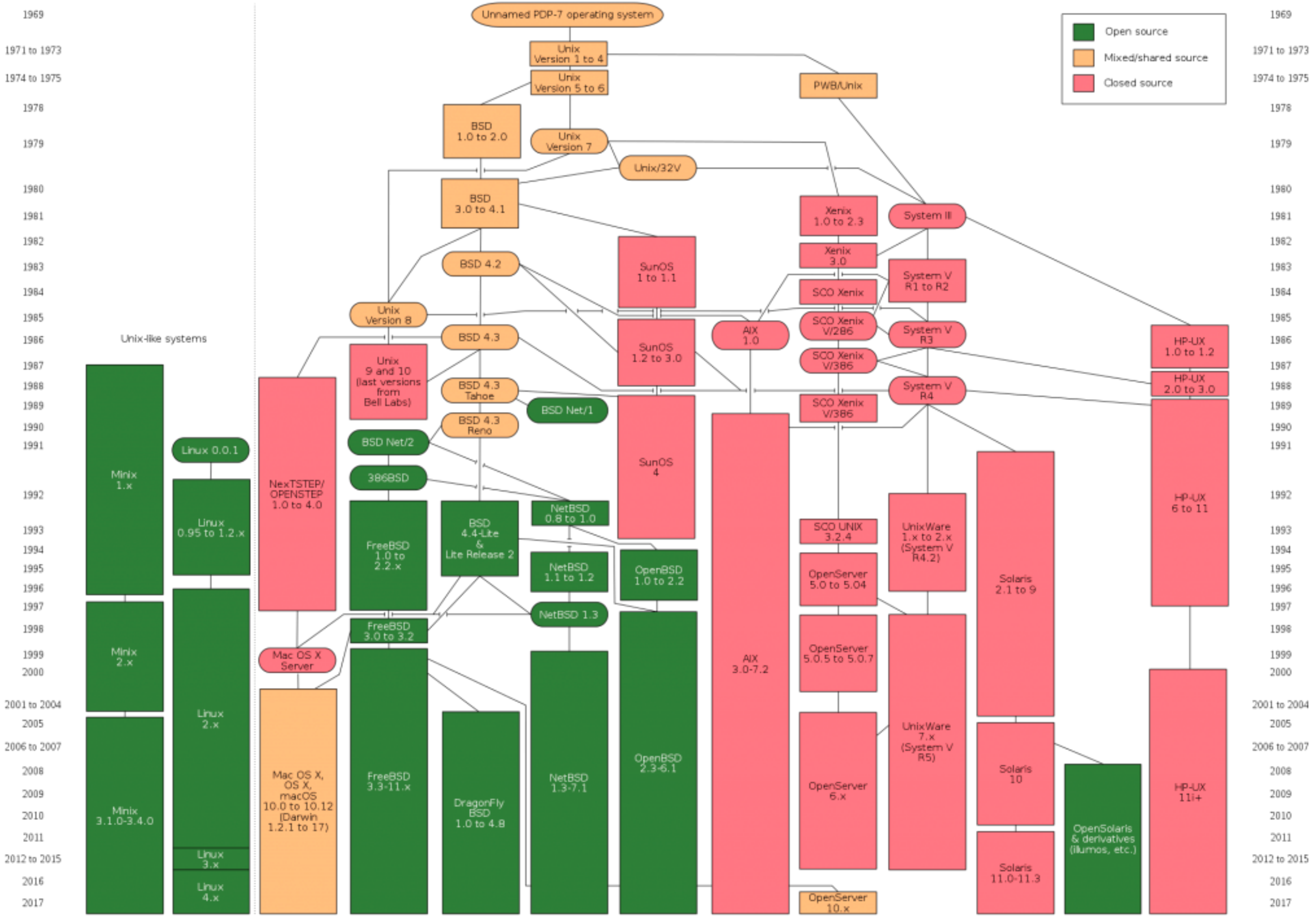
# OPERATING SYSTEMS

## Windows Over 20 Years

- Windows has generally seen many different Graphical User Interfaces (or GUI's) over the years. Generally, the advent of Windows and key innovation was to remove the need for shell level computing. However a lot of scripting and the ability to manipulate, wrangle, edit, and work with text was lost. Bioinformatics tends to need these tools, and these are generally found in the various flavors 'nix. There are ways to get to a 'nix environment within Windows such as through Cygwin or other VM devices but we generally don't discuss that in this material.

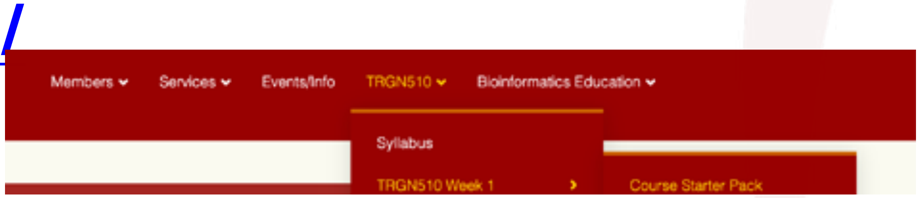


# What is Unix/Linux/Bash/Command-line/Terminal?

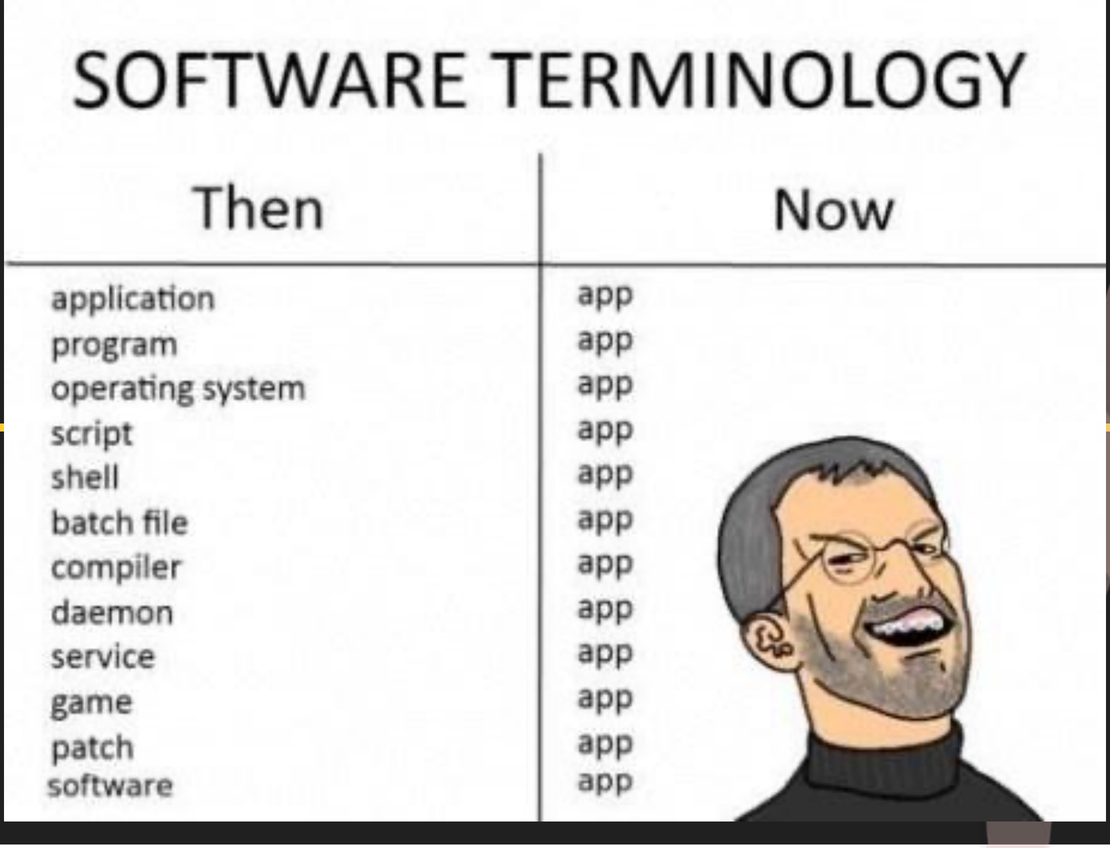


# MacOS Setup

[https://itg.usc.edu/](https://itg.usc.edu/)



# Jargon Is Just That



## SOFTWARE TERMINOLOGY

| Then | Now |
|---|---|
| application | app |
| program | app |
| operating system | app |
| script | app |
| shell | app |
| batch file | app |
| compiler | app |
| daemon | app |
| service | app |
| game | app |
| patch | app |
| software | app |

# COMMAND-LINE SHELLS

## Key Commands

- Navigating Directories

## Special Characters

- ~ / * | > < . ..

## Pipe Redirect

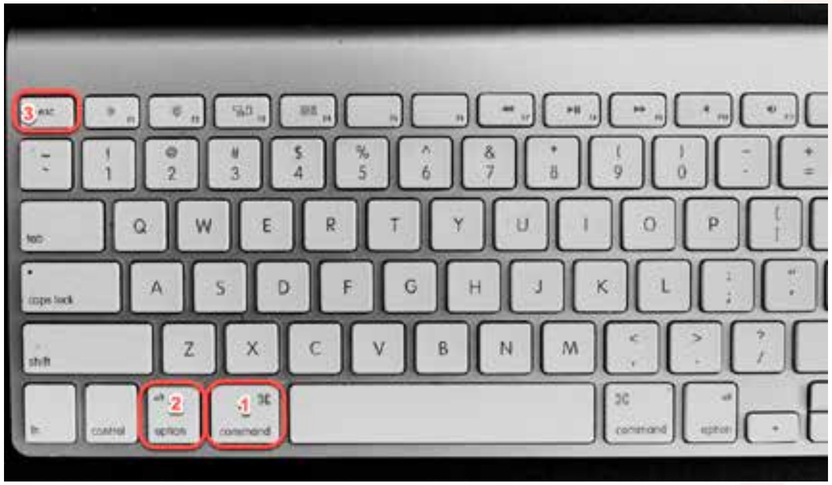
- | >

## Core commands

- Find ls , among aothers

# VIM

## A right of passage.  Why Mac brought back the “esc” character.



# Expectations

## We will cover a lot of concepts at low depth

- You will learn & demonstrate basics of 6 scripting/database/programming languages
  - Bash
  - Python
  - R
  - Javascript
  - HTML/CSS
  - SQL
- You will learn project management and computing
- You will learn about deploying servers, high-performance computing, cloud computing.
- You will develop a multi-faceted web-application using these languages as a final project

## This is a graduate level course

- You will learn to learn independently; Moving forward in the face of ambiguity is the single most important thing to learn.

## The amount of weekly time Varies By Experience.

- Average 20 hours per week, with many weeks 25 hours
  - Some people report 3 to 4 hours per week; Some report more
- Graduate level courses
  - Moving forward in the face of ambiguity
- Graduate level courses: This course has extensive material provided, but you are expected to go beyond and seek external resources too!
- The material has purposeful gaps; Learning to move forward in the face of ambiguity is key
- You want to walk away with confidence at tackling most Data Science problems
- We want to make you dangerously good at biomedical informatics by the end of the course.

# EXPECTATIONS II

## You are expected to become independent

- This is a graduate level course.
- You'll experience frustration.
- I expect you to spend at least 4 and up to 8 hours before reaching out to help. I will not respond immediately.
- I will not respond if a deadline is less then 6 hours away.
  - *Starting an assignment with 2 hours to go, and then sending multiple emails after 1 hour demanding help does not facilitate the type of learning we are seeking.*
- Do not spend more than 8 hours solving a problem without reaching to me

## Experience from Prior Years

- Those who were given the answer quickly, complained they felt they didn't learn enough in course.
- Those who received less feedback and more challenging exercised considered this to be one of most important courses they've taken.

## Engagement with servers & resources is tracked

- Understanding privacy & computers and the internet is key to the course.
- The internet does not give privacy, expect that any resource is tracking your every interaction.

## Seeking help

- Do use screen snapshots & videos
- Do reach out to colleagues

## Giving Help

- Do help others; but don't handicap them to depend on you.
- Their goal is to find resources to learn – like your own; Is your goal to be their quick answer resource?
- Do share helpful learning resources with others
- Do share fun learning resources with others.
- Google. Can't emphasize that the secret to programming is here.
- https://www.youtube.com/watch?v=HluANRwPyNo

# Resources

## Blackboard

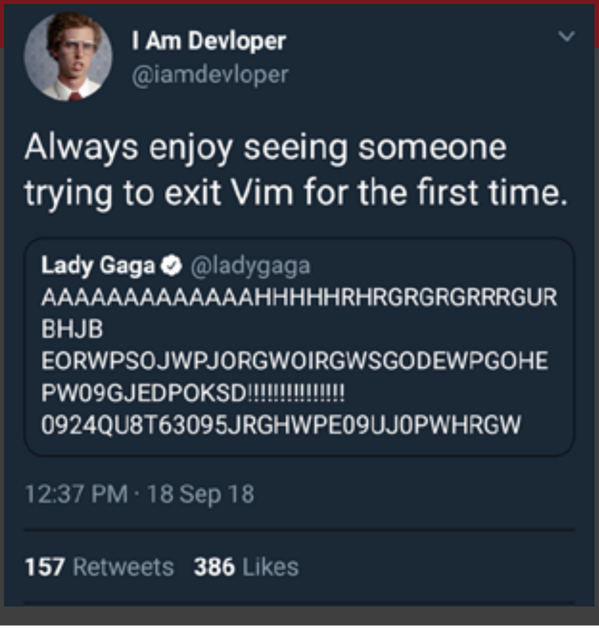
## Class Webpage

- https://itg.usc.edu/site/index.php/trgn510/

# Advanced Programming

r/programminghumor

# Programming Needs Are Often Simple



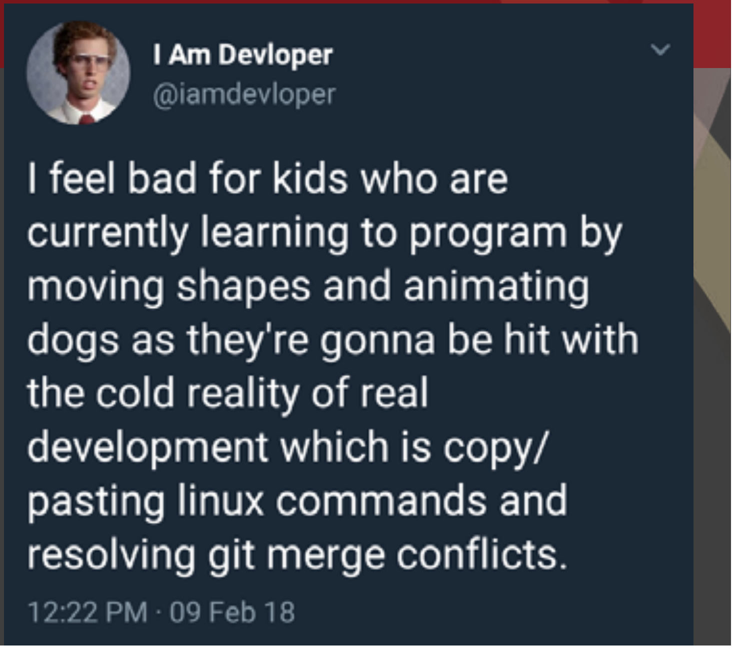
**I Am Devloper**
@iamdevloper

Always enjoy seeing someone trying to exit Vim for the first time.

**Lady Gaga** @ladygaga
AAAAAAAAAAAAAHHHHHRHRGRGRGRRRGUR
BHJB
EORWPSOJWPJORGWOIRGWSGODEWPGOHE
PW09GJEDPOKSD!!!!!!!!!!!!!!
0924QU8T63095JRGHWPE09UJ0PWHRGW

12:37 PM · 18 Sep 18

**157** Retweets **386** Likes



**I Am Devloper**
@iamdevloper

I feel bad for kids who are currently learning to program by moving shapes and animating dogs as they're gonna be hit with the cold reality of real development which is copy/pasting linux commands and resolving git merge conflicts.

12:22 PM · 09 Feb 18

# We Have Covered A Lot

# BIOINFORMATICS

- Translational bioinformatics (TBI) is an emerging field in the study of health informatics, focused on the convergence of molecular bioinformatics, biostatistics, statistical genetics and clinical informatics. Its focus is on applying informatics methodology to the increasing amount of biomedical and genomic data to formulate knowledge and medical tools, which can be utilized by scientists, clinicians, and patients.[1] Furthermore, it involves applying biomedical research to improve human health through the use of computer-based information system