Keck School of Medicine of USC

TRGN 510
Basic Foundations in Translational Biomedical Informatics

Units:	4
Term:	Fall
Date/Time:	1:00-2:50 PM Tue, Thur
Location:	NRT 2508

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

Instructor:	David W. Craig, Ph.D.
Office:	NRT 2517K
Hours:	By Appointment
Contact Info:	David W. Craig, Ph.D. Professor, Dept. Of Translational Genomics Harlyne J. Norris Cancer Research Tower 1450 Biggy St. NRT 4506 Los Angeles, CA 90089-9601 Email: davidwcr@usc.edu Office: (323) 442-7784

Course Description

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.

Prerequisite(s):

- concepts. Bachelor's degree in biology, healthcare, or chemistry.

- This course is taught as part of a Masters In Translational Biomedical Informatics, and having prior taken 514 or being concurrently enrolled in this course is a requirement

Course Notes

This course will be a blended course where students are expected to review pre-prepared lectures online and participate in group activities and homework on centralized servers. There will be in person testing and in person presentations at the Keck School of Medicine requiring attendance unless alternative arrangements are made. Students will learn by example within cloud-based servers designed to mirroring bioinformatics work-environments spanning research labs, to industry, and to clinical laboratories. To ensure a focus on the informatics challenges students will all have similar computational setups, allowing them the ability to better collaborate on problem solving and ensuring that one student's issue isn't an unusual setup of their particular environment. Students will interact through a combination of Blackboard, Slack, and Bluejeans. Each class will have a GitHub environment where assignments are available for teams and for the class. Students will be expected to be able to have access to computers and access to internet as specified below.

Technological Proficiency and Hardware/Software Required

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.

Supplementary Materials

Weekly required readings will be provided and are described in the course syllabus. Material will be pulled from biomedical journals such as Nature, Science, Cell and other top tier journals available to students via the USC Library services. Most required material is generally available at no additional costs to the student, respecting the appropriate content license.

- Code Academy.
  - Learn the Command Line. https://www.codecademy.com/learn/learn-the-command-line
  - Deploy a website. https://www.codecademy.com/learn/
  - Learn Git. https://www.codecademy.com/learn/learn-git
- Kenneth Bradnam's Command line bootcamp. http://rik.smith-unna.com/command_line_bootcamp Creative Commons License 3.0
- Little Book of R for Bioinformatics by Avril Coghlan, Wellcome Trust Sanger Institute, Cambridge. Obtain via https://a-little-book-of-r-for-bioinformatics.readthedocs.io/ Creative Commons License 3.0
- R-Shiny tutorial. https://shiny.rstudio.com/tutorial GPL 3.0 License
- DataCamp. Introduction to R. https://www.datacamp.com/courses/free-introduction-to-r
- Bash Scripting Tutorial. http://ryanstutorials.net/bash-scripting-tutorial/
- USC Center for High Performance Computing Documentation guide: https://hpcc.usc.edu/support/documentation/

Description and Assessment 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 Breakdown

- *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.

Instructor availability and Additional Help

- The structure of the course will allow for in person help between 1-2:50PM Tuesdays and Thursdays with changes being posted at the course website http://dtg.usc.edu/tgrn510 and Blackboard by scheduling time at least 48 hours in advance by emailing davidwcr@usc.edu. Due to the restricted nature of the 2nd floor of NRT, drop-in appointments are not a possibility outside of these times or without scheduling ahead of time.
- The nature of the course is to be one that fosters independence and learning, and students are guided to identify best paths for solving a problem understanding that this is a time intensive process often involving struggling with errors and uncertainty which native to this field. Students are encouraged to utilize community resources for solving problems.

Week By Week

- **August 18/20**
  - Introduction to course objectives/requirements, teaching approaches, and structure
    - Installation of key software and setup of individual work environments.
  - *Project Management and Versioning.*
    - Basics of project planning, documentation, communication, specification creation, unit testing, and project management. Introduction to Markdowns, agile programming, bug-tracking, templating.
  - *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
- **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
- **Oct 13/15**
  - Bioconductor

    - Homework Week 8
    - Quiz Week 8
- **Oct 20/22**
    - Introduction of final projects. Introduction to bioinformatics applications with a focus on genomics.
    - Building a pipeline for analysis of a human genome example
    - Homework Week 9
- **Oct 27/29**
    - Building a pipeline and reporting framework using analysis of human genome example
    - Homework Week 10
    - Quiz Week 10
- **Nov 3/5**
    - Practical Examples
    - Homework Week 11
- **November 9-13**.
    - Final Projects Due.
- **Nov 17th week**
    - Final Exam

Statement on Academic Conduct and Support Systems

Academic Conduct

Plagiarism – presenting someone else's ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in SCampus in Section 11, Behavior Violating University Standards https://scampus.usc.edu/1100-behavior-violating-university-standards-and-appropriate-sanctions. Other forms of academic dishonesty are equally unacceptable. See additional information in SCampus and university policies on scientific misconduct, http://policy.usc.edu/scientific-misconduct.

Discrimination, sexual assault, and harassment are not tolerated by the university. You are encouraged to report any incidents to the Office of Equity and Diversity http://equity.usc.edu or to the Department of Public Safety http://adminopsnet.usc.edu/department/department-public-safety. This is important for the safety of the whole USC community. Another member of the university community – such as a friend, classmate, advisor, or faculty member – can help initiate the report, or can initiate the report on behalf of another person. The Center for Women and Men http://www.usc.edu/student-affairs/cwm/ provides 24/7 confidential support, and the sexual assault resource center webpage http://sarc.usc.edu describes reporting options and other resources.

Support Systems

A number of USC's schools provide support for students who need help with scholarly writing. Check with your advisor or program staff to find out more. Students whose primary language is not English should check with the American Language Institute http://dornsife.usc.edu/ali, which sponsors courses and workshops specifically for international graduate students. The Office of Disability Services and Programs http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html provides certification for students with disabilities and helps arrange the relevant accommodations. If an officially declared emergency makes travel to campus infeasible, USC Emergency Information http://emergency.usc.edu will provide safety and other updates, including ways in which instruction will be continued by means of blackboard, teleconferencing, and other technology.