BIOS 30318/60318 - Introduction to Biocomputing
Fall 2017
10:30-11:20 MW lecture – 109 Pasquerilla
10:30-11:20 F tutorial – 317, 318, 319 DeBartolo
3 credits

INSTRUCTOR
Stuart Jones, office: 264 Galvin, email: sjones20@nd.edu; phone: 631-5703

GENERAL DESCRIPTION
    Modern biology, as well as biochemistry and biophysics, relies significantly on computation. The volumes of data generated by modern lab and field research commonly require greater capacity and more sophisticated algorithms for reformatting, filtering, and analyzing than are available in traditional spreadsheet software. As a result, an efficient and productive scientist must possess, at least basic, biocomputational skills. Often these requisite skills include the ability to navigate the Unix Shell environment, to understand and implement existing software tools, and to use a scripting language (e.g. R or Python) for data processing and analysis. This course will provide students with the knowledge and experience required to apply these important tools in diverse contexts, including statistics, modeling, and bioinformatics.
    Class meetings on Monday and Wednesday will follow an interactive lecture format. There is no textbook for this course. Rather, preparation for lectures will include readings and/or tutorial exercises, many of which are drawn from the popular Software Carpentry program. Many lectures will include a brief quiz based upon the readings or tutorials. Friday meetings will follow a tutorial format where students work individually or in groups to complete hands-on biocomputational exercises. These activities will draw on theory and concepts covered in the preceding lectures. Tutorial activities will also be graded weekly. The course will culminate in a group biocomputation project and "open-book" final exam.

LEARNING GOALS
    1. Navigate file structures and conduct basic file manipulation in the shell environment, including shell scripting

    2. Demonstrate the ability to use existing and custom designed functions to manipulate diverse data types in an automated and repeatable manner with R or Python.

    3. Implement computational best practices, including problem definition and decomposition, debugging, documentation and version control.

    4. Gain an appreciation for the central role of computation in modern science.

OFFICE HOURS
Office hours will be held Tuesday 10 – 11 and Wednesday 11:30 – 12:30. If these times don't work, email me to schedule an alternative time.

MATERIAL TO BE COVERED

Week	Topic
August 21	Shell – basics
August 28	Shell – programming
September 4	Shell – scripts
September 11	Version control
September 18	Scripting – basics
September 25	Documentation
October 2	Scripting – Control Flow
October 9	Scripting - Regex
October 16	FALL BREAK
October 23	Review and Best Practices
October 30	Packages and Graphing
November 6	Problem Decomposition, Debugging, and Defensive Coding
November 13	Applications: Statistics, Models, and Bioinformatics
November 20	Introduction to the Center for Research Computing
November 27	Projects
December 4	Projects

ASSESSMENTS

Quizzes –10% of grade
Approximately 15 quizzes will be administered at the beginning of lecture across the semester. These will be based on readings and flipped-classroom work done. These quizzes ensure students are prepared for class meetings and provide "low-risk" evaluations of student comprehension.

Good & Bad – 15% of grade
Each Wednesday of the semester students will be asked to provide responses to two questions: What concept remains a bit unclear to you? and What did the instructors do that most facilitated your learning this week?. Responses will not be graded, but students will receive credit for completing the exercise each week. These responses will allow for adaptive review of material in the tutorial on Friday each week.

Graded Exercises – 30% of grade
Each tutorial will consist of a hands-on activity that will include short-answer questions to be completed by the students. Responses to these questions or some other output of the activity (e.g. a script on GitHub) will be graded. These exercises will be due by the beginning of class on the following Friday.

Group Project – 30% of grade
We will use the last two weeks of class to develop and complete a group project. Prior to this activity, the students will work through a real world application of scientific computing in the general areas of statistics, modeling, and bioinformatics. Based on this experience students will select a topic for their group computational project. Code developed collaboratively on GitHub will be graded. Overall quality of the product will make up most of the grade, but individual contributions to the

project will also be tracked via "issues" and "commits" on the GitHub repository. We will use the last day of the semester to present group project results.

Final Exam – 15% of grade
A final exam will be administered from 4:15 to 6:15 on Tuesday, December 12th in 109 Pasquerilla. The exam will be used to assess each student's progress towards the learning goals listed above, and therefore will be completed individually. Because the learning goals emphasize application rather than content, the exam will be "open book".

ACADEMIC INTEGRITY
Plagiarism will not be tolerated per the Notre Dame Academic Code of Honor. Consequences will be severe and all cases will be reported to the Dean for Undergraduate Studies office. If you need clarification on what constitutes plagiarism see http://www.library.nd.edu/help/plagiarism.shtml

DETAILED SCHEDULE
Week 1
Wednesday, August 23
    -Pre: NA
    -Material: Introduction to course structure, a bit about The Shell, and why learn programming
    -Post: Why you want to program?

Friday, August 25 - TUTORIAL
    -Pre: SWC Unix Shell – Introducing the Shell
    -Material: SWC Unix Shell – Navigating Files and Directories, Working with Files and Directories
    -Post: **Graded exercise**

Week 2
Monday, August 28
    -Pre: SWC Unix Shell – Pipes and Filters; **QUIZ**
    -Material: Concepts of pipes and filters; examples with sort, head, tail, and tr
    -Post: NA

Wednesday, August 30
    -Pre: SWC Unix Shell – Finding Things; **QUIZ**
    -Material: Concepts and examples with awk and grep
    -Post: **Good & Bad**

Friday, September 1 - TUTORIAL
    -Pre: NA
    -Material: Shell pipes and filters exercise
    -Post: **Graded exercise**

Week 3
Monday, September 4

-Pre: SWC Unix Shell – Loops; **QUIZ**
-Material: Concepts and use cases for loops, includes discussion of shell variables
-Post: NA

Wednesday, September 6
-Pre: SWC Unix Shell – Shell Scripts; **QUIZ**
-Material: Concepts and examples of shell scripts
-Post: **Good & Bad**

Friday, September 8 - TUTORIAL
-Pre: SWC Version Control with Git - Introduction
-Material: SWC Version Control with Git – Setting up, making directory, tracking changes; Exercise creating a version-controlled shell script
-Post: **Graded exercise**

Week 4
Monday, September 11
-Pre: Reading about Version Control and Git; **QUIZ**
-Material: Git basics review and expand
-Post: NA

Wednesday, September 13
-Pre: Sign up for GitHub
-Material: Github theory and usage
-Post: **Good & Bad**

Friday, September 15 - TUTORIAL
-Pre: SWC Version Control with Git – Remotes in Github
-Material: SWC Version Control with Git – Collaborating, Conflicts
-Post: **Graded Github exercise**: clone from TA, edit in small group, pull request, resolve conflicts, etc.

Week 5
Monday, September 18
-Pre: Reading; **QUIZ**
-Material: Scripting languages – usage, options, etc.
-Post: Pick between R and Python

Wednesday, September 20
-Pre: NA
-Material: Scripting – variables, I/O, functions & arguments, data structures
-Post: **Good & Bad** + define data structures from preferred language in own words

Friday, September 22 - TUTORIAL
-Pre: NA
-Material: SWC Programming – Patient Data
-Post: **Graded Patient Data Exercise**

Week 6
Monday, September 25
    -Pre: Reading; **QUIZ**
    -Material: Documentation concepts and applications
    -Post: NA

Wednesday, September 27
    -Pre: SWC Programming – Documentation
    -Material: ipython/knitr & markdown
    -Post: **Good & Bad**

Friday, September 29 - TUTORIAL
    -Pre: NA
    -Material: Group exercise with Github, R/Python, and Markdown
    -Post: **Graded exercise from Github repos**

Week 7
Monday, October 2
    -Pre: SWC Programming – Making Choices; **QUIZ**
    -Material: Making Choices – theory and application
    -Post: NA

Wednesday, October 4
    -Pre: SWC Programming – Repeating actions with loops
    -Material: Analyzing multiple sets with loops – theory and application
    -Post: **Good & Bad**

Friday, October 6 - TUTORIAL
    -Pre: NA
    -Material: Exercise applying loops and choices with questions
    -Post: **Graded exercise**

Week 8
Monday, October 9
    -Pre: Reading; **QUIZ**
    -Material: Regular expressions theory
    -Post: NA

Wednesday, October 11
    -Pre: NA
    -Material: String manipulations
    -Post: **Good & Bad**

Friday, October 13 - TUTORIAL
    -Pre: NA
    -Material: String manipulation exercise, including using R/Python in command line/shell script
    -Post: **Graded exercise**

Week 9
Monday, October 23
    -Pre: Personal map of learning across semester
    -Material: Discussion of where we've been in semester
    -Post: NA

Wednesday, October 25
    -Pre: Think about assigned programming scenarios and best practices
    -Material: Discuss scenarios and best practices
    -Post: NA

Friday, October 27 - TUTORIAL
    -Pre: Strategize about scenarios as team using markdown docs or issues on github
    -Material: Mini-implementation of scenarios and best practices
    -Post: **Graded group exercise**

Week 10
Monday, October 30
    -Pre: Reading; **QUIZ**
    -Material: Package repositories with graphics as example
    -Post: NA

Wednesday, November 1
    -Pre: Reading
    -Material: Graphics visualization theory and examples
    -Post: **Good & Bad**

Friday, November 3 - TUTORIAL
    -Pre: NA
    -Material: Graphics exercise
    -Post: **Graded exercise**

Week 11
Monday, November 6
    -Pre: NA
    -Material: Problem decomposition discussion and practice
    -Post: NA

Wednesday, November 8
    -Pre: Reading; **QUIZ**
    -Material: Debugging and defensive coding
    -Post: NA

Friday, November 10 - TUTORIAL
    -Pre: NA
    -Material: Debugging and defensive coding exercise
    -Post: **Graded exercise**

Week 12
Monday, November 13
   -Pre: Reading
   -Material: Statistics application
   -Post: **Graded questions**

Wednesday, November 15
   -Pre: Reading
   -Material: Modeling application
   -Post: **Graded questions**

Friday, November 17
   -Pre: Reading
   -Material: Bioinformatics application
   -Post: **Graded questions**

Week 13
Monday, November 20
   -Pre: Reading from CRC wiki; **QUIZ**
   -Material: Introduction to Center for Research Computing
   -Post: NA

Week 14
Monday, November 27
   -Pre: Pick a project topic
   -Material: Project problem decomposition
   -Post: NA

Wednesday, November 29
   -Pre: Project plan documented
   -Material: Project work
   -Post: NA

Friday, December 1
   -Pre: NA
   -Material: Project work
   -Post: NA

Week 15
Monday, December 4
   -Pre: NA
   -Material: Project work
   -Post: NA

Wednesday, December 6
   -Pre: NA
   -Material: **Project presentations**
   -Post: NA