



How to Present a Code-Based Course to Students with No Previous Coding Experience



David Reif, PhD

Associate Professor

North Carolina State University

Raleigh, NC

www.reif-lab.org



Agenda

Summarize experience in developing a semester-length course on practical computing

– GN427: Introduction to Bioinformatics

- Advertising and attracting students
- Learning outcomes (objectives)
- Course design
- Example of a classroom session
- Student outcomes and feedback
- Resources



Advertising and Attracting Students

- Don't scare them
 - Flyer explicitly states: “Previous coding experience is NOT necessary”
- Show something they'd want to know how to do
- You might tap a surprising need/demographic:
 - 1st three offerings of GN427 (undergrad listing) had >50% enrollment of grad/post-grad students
- Prepare for many enrollment-override requests...

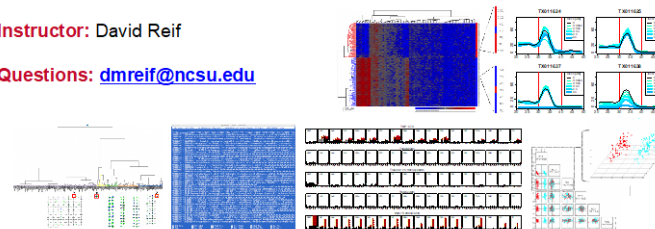
New Course for Spring 2014

GN427: Introduction to Bioinformatics

Class Meetings: Tuesday / Thursday @ 10:15 - 11:30

Instructor: David Reif

Questions: dmreif@ncsu.edu



Functional Programming Framework

The course will provide a foundation in biological computing. Students will learn to code in R (largely platform-independent) and the basics of operating in a Unix environment. Previous coding experience is NOT necessary.

Data Visualization

Visualization will be emphasized as an effective means of interdisciplinary communication.

Data Manipulation

Students will learn the basics of interacting with data that are too big to view in spreadsheet software (e.g. Excel).

Bioinformatics Applications

The class will focus on the many important bioinformatics tools/applications that are accessible with basic coding knowledge.



Course Learning Outcomes

By the end of this course, students will be able to:

- Write and debug computer programs in R.
- Create, extract, and reformat large data sets.
- Create and edit graphics for genetic and biological data.
- Automate analyses with scripts.



Course Design

Grading is setup to promote “daily coding”

- 50% Homework: Mini-projects requiring formal reports (graphics + text) and source code
- 20% Exercises: Activities (in-class) or Assignments (short homework)
- 15% Mid-term Exam: “open-world” of resources to test problem solving
- 15% Final Exam: Cumulative test of ability to use resources for problem solving



Example of a Classroom Session

Each session is 75 minutes in a classroom where each student has their own desktop (or personal laptop if they prefer) and includes the following:

- Introduce topic and provide context on why they should care
- Active learning 1: Students find info on open-ended question
- Code examples: Students follow-along with posted code
- Active learning 2: Small-group exercise applying lecture code
- Recap: What did we learn? How can I apply this?
- Assignments (short, daily exercises) or Homework (projects)
- Time for individual questions/help



Example of a Classroom Session

GN 427: Introductory Bioinformatics

Fall 2017

◆ 25. Bioinformatics applications: Alternative (contributed) packages

Introduce topic →

- a. Bioinformatics applications with *Biostrings*: Alternative package for biological data handling and manipulation
 - i. Provides useful data structures and functions for working with common types of data encountered in modern Bioinformatics
 - ii. Compare the operations here to those using base R text-matching functions
 - iii. This package is useful for manipulating data for downstream analysis (either by you or passed to a collaborator)

Active learning 1 →

- b. **Activity**: Count the number of occurrences of the pattern “GC” in the sequence of the human ABCA1 gene.

Code examples →

- c. Example lecture code using the *Biostrings* package

Active learning 2 →

- d. **Activity**: Using the human ABCA1 gene in FASTA format (https://www.ncbi.nlm.nih.gov/nuccore/NC_000009.12?report=fasta&from=104781002&to=104928246&strand=true), count the number of occurrences of the pattern “GC” in the sequence. Show the lines of R code that you used to obtain your count, plus the value itself (i.e. the number of occurrences).

Homework →

- e. **HW5 Posted**: Due November 30th



Student Outcomes and Feedback

Outcomes

- Students can generate code to solve real-world bioinformatics problems (Example below)

Question 1.6

Upon translation of the first 999 bases of the MTHFR sequence, which of the resulting amino acids occur at least 30 times?

The amino acids that occur at least 30 times are Alanine (A), Arginine (R), Leucine (L), Glycine (G), and Proline (P).

```
littles <- subseq(myseq, start = 1, end = 999)
transseq <- translate(littles)
freqaa <- alphabetFrequency(transseq)
```

```
which(freqaa > 29)
```

```
## [1] 1 2 8 11 15
```

```
freqaa[which(freqaa > 29)]
```

```
## [1] 30 34 41 32 36
```

```
colnames(freqaa)[which(freqaa > 29)]
```

```
## [1] "A" "R" "G" "L" "P"
```




Student Outcomes and Feedback

Feedback

- “I learned during the exam!”
- “Daily assignments helped make me get-to-know my computer and get comfortable with coding.”
- “I actually had fun working on the homework.”
- “I used my new skills to generate graphics for a post-doc in our lab.”
- “Now I want to go to grad school for computational biology.”
- “I want to stay at the bench but will use these skills.”



Resources

There are a wealth of free, online resources for learning R, as well as an active user community sharing solutions to posted questions

- CRAN (<https://cran.r-project.org/>)
- Bioconductor (<https://www.bioconductor.org/>)
- Online courses
 - edX “Introduction to R for Data Science”
(<https://www.edx.org/course/introduction-r-data-science-microsoft-dat204x-7>)
 - Data Camp “Introduction to R”
(<https://www.datacamp.com/courses/free-introduction-to-r>)
- Tutorials
 - R Studio: Popular integrated development environment (IDE) for R
<https://www.rstudio.com/online-learning/>