Syllabus: Computational Biology (BIOL F694)

January 18, 2006

1 Instructor contact information

Naoki Takebayashi, WRRB 226, 474-1178
e-mail: flint@uaf.edu
Office hour: any time but you can drop me an e-mail before you come
E-mail is the best way to contact me.

2 Meeting time and place

TR, 9:45-11:15AM, Irving I 303 3 credits

3 Course Description

Computation has been used in biology since 1960’s. In the recent years, computational biology has been
moved into the central domain of biological science. The boom in computational biology is motivated by
both availability of enormous data set (bioinformatics) and complexity of biological systems (simulational
studies). Programming skill is essential in computational biology, but it may not be readily accessible to
most biologists without guidance. Yet, practical programming skills needed for biological problems are
relatively simple. It can be learnt and applied for their own biological research without formal computer
science courses. The course will expose students to the first-hand experience of programming, specifically
tailored for biological applications. The goal of the course is that students without any previous
programming experiences become able to apply the programming skills to solving their daily biological
problems.

This course is motivated by my personal experiences helping my friends, who were painfully analyzing
huge data sets by hand (with spreadsheet programs) or whose brilliant idea could not be tested, because
they did not know how to do computer simulations.

First, we will cover basic unix environment. Then, we will cover simulational approaches with C, using
ecological and evolutionary examples. Finally, we will learn higher-level languages, such as Perl and
R, which are useful for large-scale data analysis and visualization. In each section, we will begin with biological questions, and then we will investigate how to approach the problem. The underlying theory or statistical techniques will be discussed, and programming techniques and algorithms will be explained. We will employ several programming languages (C, perl, R), which has strengths and weakness and complement each other.

The students should have elementary knowledge of computers (e.g., how to use keyboard, mouse, etc), but are not expected to know how to program or work within unix computer environment. During the class, we will meet in a computer lab, and access unix server, but students are encouraged to use your own computer.

One of the most important part of the course is working on a self-motivated programming project. The project should be related to your own interests: ideally a publishable results, or developing tools useful for your research. The students should demonstrate the competency in programing. Within the first 4 weeks, the students will hand in a brief (1-2 pages) proposal, describing the problems you are going to tackle. Each student will give a final talk at the end of the semester, and write a formal paper on the project.

4 Approximate schedules

Week 1. Unix environments:
- Unix basic commands, text-editor, setting up compiler environment.

Week 2. Basics in C
- variables, flow control, conditional, logical expression
- birth-death models
- quick R tutorial

Week 3. Basics in C
- Functions, arrays
- logistic birth-death

Week 4. Basics in C
- Pointers, Dynamic memory allocation
- genetic drift, random walk
- Student presentation of project ideas

Week 5-6. Algorithms
- random number generation, hash, sorting
- Review of Probability distribution
- Prey-predator

Week 7. Structure
- data structure
- Foraging model
Week 8-9. Application and efficiency
- optimization tips
- Maintenance of gynodioecy
- Resource allocation theory
- Spatial population dynamics

Week 10. Using libraries
- GNU scientific library (GSL)

Week 11. Perl basics
- text handling
- regular expression

Week 12. Application of perl to DNA sequence analysis
- bioperl
- interface to genbank
- sequence manipulation

Week 13-14 Visualization and statistics with R

Week 15. Student’s presentation

5 Course readings/materials

No textbook is required.

But here are a couple of useful books. No need to get these for the course, but they may be useful for your future references:


6 Course goals

Students will learn basic programming skills useful for biological problems. After the completion of the course, students should feel comfortable in developing computer simulation, or make programs for biological data analyses.
7 Instructional methods

Students will learn through lecture, reading, and group discussion.

8 Course policies

You are expected to attend lectures and participate in discussion. You are expected to arrive at lecture on time.

9 Requirements

All students will be required to do readings and homework assignments. I encourage students to work on the homework assignments together. You are likely to "feel" the real meanings of concepts or techniques by exchanging different ways of interpreting them with your colleagues.

Additionally, part of the grade is based on your programming project. I encourage you to choose a topic which is related to your own research.

One of the most important part of the course is working on a self-motivated programming project. The project should be related to your own interests; ideally a publishable results, or developing tools useful for your research. The students with similar interests can form small groups (2-3 people per group), and work together on working on the projects. The students should demonstrate the competency in programming. Within the first 4 weeks, the students will hand in a brief (1-3 pages) proposal, outlining the biological problems you are going to tackle. Before the proposals, students are encouraged to come to talk to me about your ideas of projects. Students will present the results during the final week. Then the manuscripts is due in the final week. I expect the manuscripts to be full-blown, publication quality (15-25 pages).

10 Evaluation/Grading

Student performance will be evaluated with the following factors 20% assignments and quiz
10% proposal (due week 4)
20% presentation (due final week)
40% final manuscripts (due final week)
10% participation to group discussion

11 Support Services

If you require more assistance than can be provided in class, lab and office hours, you may want to contact Student Support Services (http://www.uaf.edu/ssp/).
12 Disability Services

If you have a disability, or think you may have a disability, please contact the Office of Disabilities Services (203 WHIT, 474-7043). We will work with this office to provide reasonable and appropriate accommodation to students with disabilities.