BIOS 735: Statistical Computing

Course Description: This class teaches important concepts and skills for statistical software development using case studies. After this course, students will have a good understanding of the process of statistical software development, knowledge of existing resources for software development, and the ability to produce reliable and efficient statistical software.

Content: In this class, students will learn

- C++ language basics, Rcpp
- Software design, documentation, compiling, testing, debugging, distribution, and maintenance
- Reproducible research
- Some specific programming techniques such as recursion, enumeration, dynamic programming, etc.
- Statistical learning methods, optimization techniques and their implementation

Prerequisites:

- Biostatistics 660, 661, 662, and 663.
- One programming class at the undergraduate level or equivalent training; students without programming experience are required to take an ITS training course for Perl, Python, or Matlab.
- Basic knowledge of the Linux environment. Students without experience using Linux are suggested to take a short ITS training course for Linux.

Instructor:

Yun Li, Bahjat Qaqish, Mengjie Chen, and Wei Sun

Required Textbook(s)

Accelerated C++: Practical Programming by Example
Andrew Koenig, Barbara E. Moo
Paperback: 352 pages
Publisher: Addison-Wesley Professional; 1 edition (August 14, 2000)

Further Reading

The Practice of Programming (Addison-Wesley Professional Computing Series)
Brian W. Kernighan, Rob Pike
Paperback: 288 pages
Publisher: Addison-Wesley; 1 edition (February 4, 1999)

C++ Primer (5th Edition)
Stanley B. Lippman, Jose Lajoie, Barbara E. Moo,
Completely Rewritten for the New C++11 Standard
Paperback: 976 pages
Publisher: Addison-Wesley Professional; 5 edition (August 16, 2012)

Graded Work:
The class will be taught through several modules. For each module, grade is assigned based on quiz or assignment. There is no midterm or final exam. The final grade is assigned based on the cumulative grades of all modules, and it will use the following grading system for graduate courses in the School of Public Health:

- H: Clear excellence
- P: Entirely satisfactory
- L: Low passing
- F: Fail

The School of Public Health grading system is designed so that the mode of the grading distribution is P. The last graded assignment will be due on the last week of regular classes.

Week 1-3 (taught by Dr. Yun Li):

Lecture 1: Introduction to LINUX

LINUX overview, staring a LINUX terminal, and basic commands

Lecture 2: C++ Basics

- Identifier, variables, constants
- Operators, expressions, statements
- Main function, compiling/linking
- C++ division and type casting
- iostream for I/O
- if-else; switch
- loops
  - while loop
  - do-while loop o for loop
- Compiling and Running

Lecture 3: C and C++ String

Lecture 4: Functions

Lecture 5: Arrays

Lecture 6: Introduction to Object Oriented Programming (OOP)

Week 4-6 (taught by Dr. Bahjat Qaqish):

Week 4. A case study of software development, for example, simple tasks such as computing mean, median, and SD of a set of numbers, but sufficient to reveal the complexities inherent in software design. The focus will be on how to design, test, document and debug the code and how to evaluate its performance. Other covered topics how to read an unknown number of elements from a file, how to handle input errors, comparing several ways to compute the mean and SD, and comparing their accuracy and performance.

Week 5. Optimization method such as Newton-Raphson method, EM algorithm, etc

Week 6. Enumeration, such as permutation or bootstrapping

Week 7 (taught by Dr. Mengjie Chen):

Introduction to two libraries eigen and Armadillo, and their interface in R through RCpp.

Eigen (http://eigen.tuxfamily.org/) is a C++ template library for linear algebra: matrices, vectors, numerical solvers, and related algorithms.

Armadillo (http://arma.sourceforge.net/) is a high quality C++ linear algebra library, aiming towards a good balance between speed and ease of use; the syntax (API) is deliberately similar to Matlab

Reproducible research. Introduction to Sweave, knitR.

Week 8 (taught by Dr. Wei Sun):

Week 8. Dynamic programming and Hidden Markov Model (HMM)

- Viterbi algorithm to find the most likely path of the hidden states
- Forward-backward algorithm (an example of EM algorithm) to estimate the parameters of HMM

Week 9-15 (taught by Dr. Mengjie Chen):

Clustering and Classification

- Clustering
  - K-means clustering,
  - Hierarchical clustering
  - Self organizing maps
- Classification
  - K-nearest-neighbor classifiers
  - SVM
  - Classification tree
- Neural Networks

Convex Optimization

- Linear programming,
- Quadratic programming

MCMC, coordinate ascent, penalization methods and its Bayesian interpretation.

- Standard MCMC methods;
  - Spike and slab prior
  - Shrinkage prior
- Variable selection methods;
  - Standard lasso, elastic net, SCAD;

  - Group lasso, overlap lasso;
  - Screening methods.

GPU programming

Other topics may be covered:

- Augmented Lagrangian Method and ADMM,
- Graphical models,
- Bayesian Nonparametric methods