MSc in Bioinformatics for Health Sciences

ALG. Introduction to Algorithmics

Syllabus Information

Academic Course: 2018/19

Academic Center: 804 - Official Postgraduate Programme in Biomedicine

Study: 8045 – Bioinformatics for Health Sciences - MSc

Subject: 31031 – ALG. Introduction to Algorithmics

Credits: 5.0

Course: 1st

Teaching languages: English

Teachers: Josep Francesc Abril Ferrando

Teaching Period: 1st term

Presentation

This course will illustrate the most basic tools and programming concepts using as the main language shell scripting. Particular effort will be spent on how to use efficiently use GNU/Linux as programming environment. Basic algorithmic concepts will be also introduced, as well as how to handle data structures like trees and graphs.

Associated skills

General competences:

Learning basic concepts of computers architecture and operative systems.

Introducing fundamental concepts on algorithms and data structures.

Acquiring skills on tools and concepts required to create software.

Understanding Unix-like and open source philosophies and standards.

Specific competences:

Acquiring proficiency at user level on Linux distributions, desktop and general tools.

Introduction to the Linux programming toolset (editors, interpreters, compilers, shell environment, etc...).

Understanding the logic of interpreted and compiled programming languages.

Practical implementation of first simple scripts and control over most useful Unix commands.

Understanding general topics of software management: the development cycle, control of versions, process automation, etc…

Focus on tree and graph data structures that can be useful in Bioinformatics.

Prerequisites

No special prerequisite are needed for this course. However, some confidence with command-line interfaces will be good. Ideally the students would also have GNU/Linux installed on their computers, but we can assist them on that point at tutorial time when the lectures start.

Contents

The order of the lectures can be modified in order to improve the narrative and the connectivity of the concepts. What follows is a general schema of the main topics being taught in ALG.

Contents section 1: Concepts of Computational Science

1.1. Introduction to Operating Systems and Linux.

1.2. Introduction to Algorithms and Data Structures.

1.3. Execution Flow and Control Structures.

1.4. Algorithms on Trees and Graphs.

Contents section 2: Unix Programming Environment

2.1. An Overview to Shell Command-line.

2.2. Shell Input/Output, Files and Filesystem.

2.3. Regular Expressions: GREP and SED.

2.4. AWK: Parsing Data Records.

2.5. Shell Scripting.

Contents section 3: Concepts Applied

3.1. Algorithms, two case-based examples: Fractals and Sorting.

3.2. Genomic Analyses on *nix Shell.

3.3. Version Control and Reproducibility in Science.

3.4. Recapitulation and open discussion.

Teaching methods

The course will be focused on teaching concepts and skills that the students will be able to apply to specific or advanced programming subjects from this Master.

1. The fundamental programming tools and concepts will be presented and reviewed. These are common to most programming languages, yet the main focus will be the shell command-line interpreter and the Unix scripting languages.

2. Practical hands-on experience will be as important as theoretical knowledge, so students will be introduced during class to problems that will be solved in the computer lab, with or without the help of the teacher.

3. Proactive participation and discussion will be encouraged during practical hands-on sessions.

Concepts	Procedures	Goal
How computers and programs store information	Bytes, memory, types, data structures, devices, streams	Basic understanding of computers and software architecture
How to control the flow of a program	Flow and logical instructions (if, for, while,...)	Understand computers logic to write software
How to create a program from scratch	GNU/linux, terminal, shell, filesystem, processes, vim	Write, save and compile instructions
BASH: A quick introduction to a scripting language	Shell scripts, advanced *nix commands	Learning BASH scripting, script execution, and Unix filters

Training activities

Students are expected to follow the steps for the computational protocols described in each “hands-on” lecture, complementing the theoretical background concepts explained on the lectures. At the end of each “hands-on” session there can be exercises proposed to the student to help him/her consolidating the concepts illustrated during the session. The students will have to deliver one or two mid-term mini-reports (2 pages), describing the solution to an algorithmic problem.

Evaluation

Continued self-evaluation: Being this a course with a substantial amount of practical work done in the computer lab, the students can be assessed during hands-on exercises, where they can show their interest and ability to follow the explanations and solving the challenges in-class. A small amount of course time will be dedicated to quick assessment of students’ progress, by reviewing practices and mid-term exercises, as well as a final recapitulation by discussing the end term exam exercises.

Coursework: Students will be asked to submit one to three mid-term exercises, which will be assigned from week 3 or later. This will motivate them to spend time revising the course materials before the final exam.

Exams: There will be a final exam based on the exercises assigned during classes. The exam will consist on a written test with questions and programming exercises to evaluate applied concepts and Unix tools proficiency.

Grading system

If quizzes are used for self-evaluation:

- Self-evaluation quizzes (10%).
- Mid-term coursework (20%).

Otherwise, if no self-evaluation quizzes are used:

- Mid-term coursework (30%).

Final exam (70%).A minimum final performance of 50% is required to pass the subject.

Bibliography and Information Resources

Most of the references will be listed at the subject website, here we list a selection of useful reference books describing related topics:

- D-A LeBlanc & R Blum
  "Linux For Dummies"
  8th Ed, 2007, John Wiley & Sons.
- E Siever, A Weber, S Figgins, R Love & A Robbins
  "Linux in a Nutshell"
  5th Ed, 2005, O'Reilly Media Inc.
- L Null & J Lobur
  "The Essentials of Computer Organization and Architecture"
  4th Ed, 2015, Jones & Bartlett Learning.
- GT Heineman, G Pollice, S Selkow
  "Algorithms in a Nutshell"
  2008, O'Reilly Media Inc.
- H Cormen, CE Leiserson, RL Rivest, & C Stein
  "Introduction to Algorithms"
  2nd Ed, 2001, The MIT Press.
- NC Jones & PA Pevzner
  "An Introduction to Bioinformatics Algorithms"
  2004, The MIT Press.
- J Orwant, J Hietaniemi & J Macdonald
  "Mastering Algorithms with Perl"
  1st Ed, 1999, O'Reilly Media Inc.
- C Newham & B Rosenblatt
  "Learning the bash Shell"
  3rd Ed, 2005, O'Reilly Media Inc.
- C Albing, JP Vossen & C Newham
  "bash Cookbook"
  1st Ed, 2007, O'Reilly Media Inc.
- JEF Friedl
  "Mastering Regular Expressions"
  3rd Ed, 2006, O'Reilly Media Inc.
- D Dougherty & A Robbins
  "sed & awk"
  2nd Ed, 1997, O'Reilly
- K Bradnam & I Korf
  "Unix and Perl to the Rescue!"
  2012, Cambridge University Press.
- C Gibas, P Jambeck
  "Developing Bioinformatics Computer Skills"
  2001, O'Reilly Media Inc.