Programming Language for Bioinformatics
生物计算编程语言

Chaochun Wei

Shanghai Jiao Tong University

Spring 2014

Contents

- Programming languages for Bioinformatics
- Course information
  - Goal
  - Contents
  - Organization
  - Grading

How many programming languages you know?

C

C++

Pascal

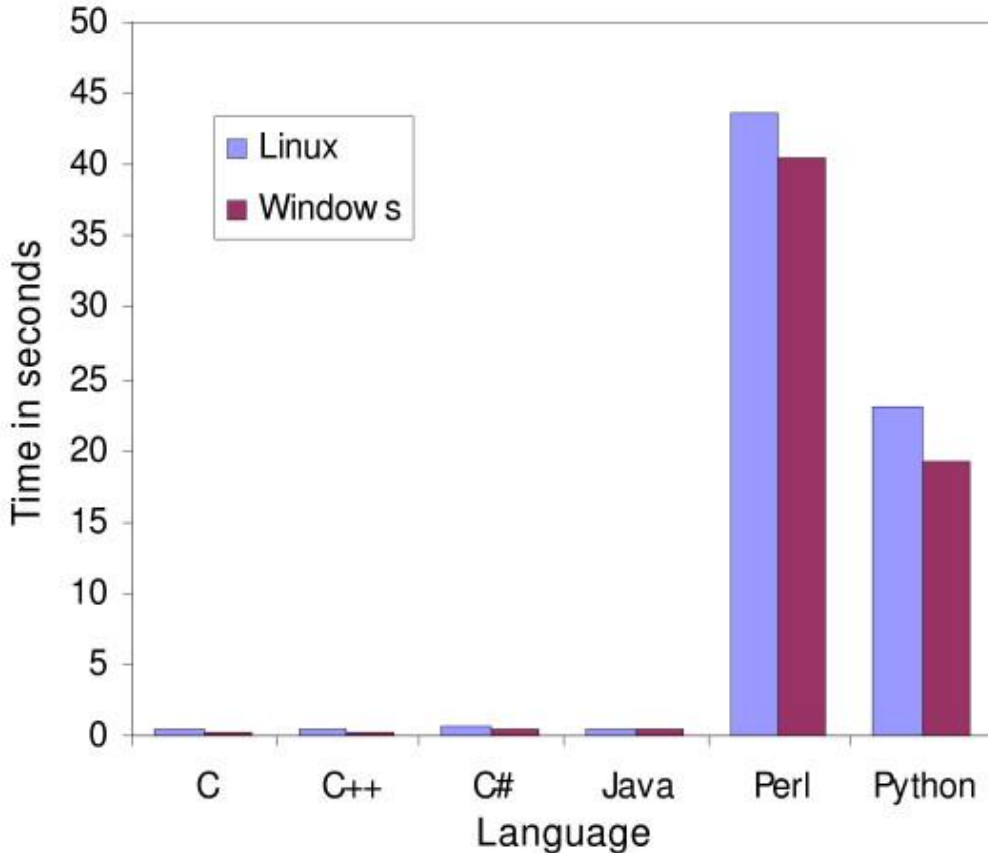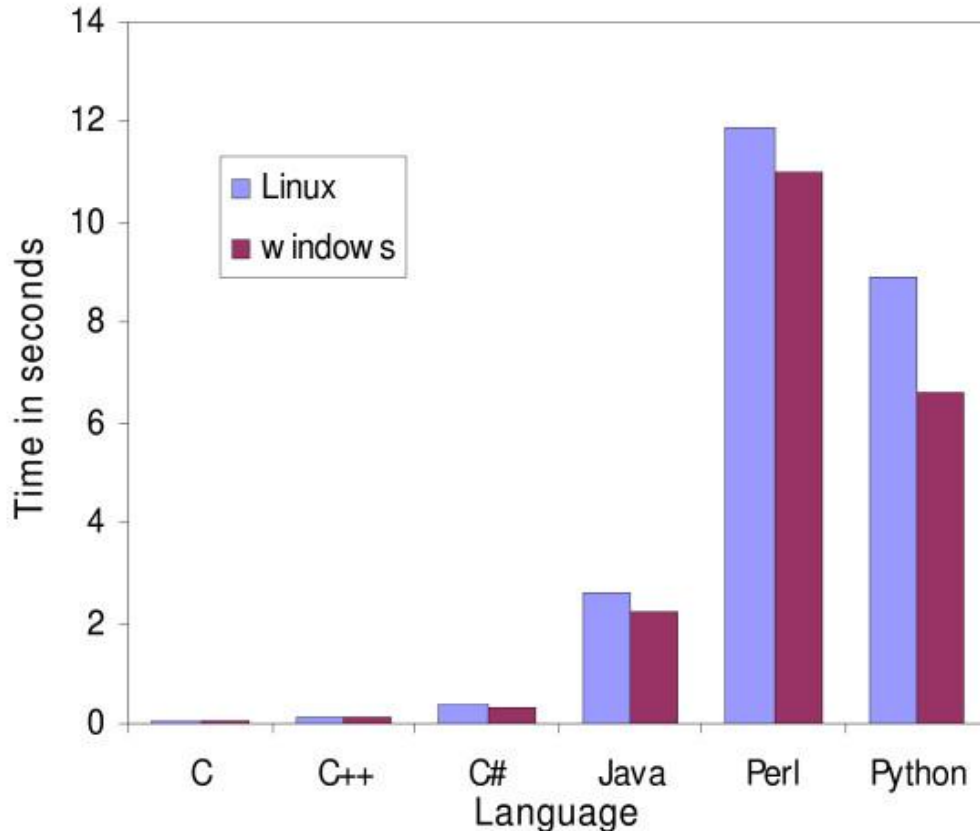Java

Basic

Perl
Python
Rubby
. . .

Matlab
Mathematica
R

■ ■ ■ ■ ■ ■

Speed comparison of the global alignment program. Speed comparison of the global alignment algorithm using a gap penalty of 10 implemented in C, C++, C#, Java, Perl and Python. The programs were run on Linux and Windows platforms. Two DNA sequences of 3216 bp and 3217 bp were used.
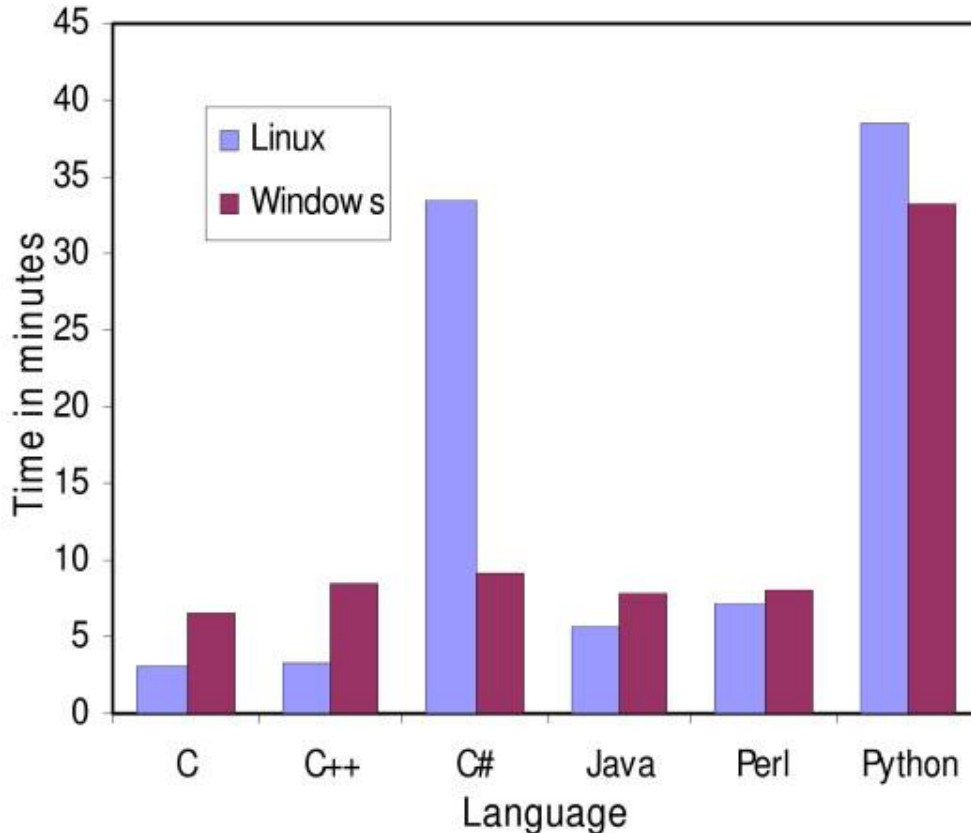
BMC Bioinformatics. 2008; 9: 82.

Speed comparison of the Neighbor-Joining program. Speed comparison of the Neighbor-Joining algorithm using the Jukes-Cantor evolutionary model implemented in C, C++, C#, Java, Perl and Python. The programs were run on Linux and Windows platforms. The input file was an alignment of 76 DNA sequences.
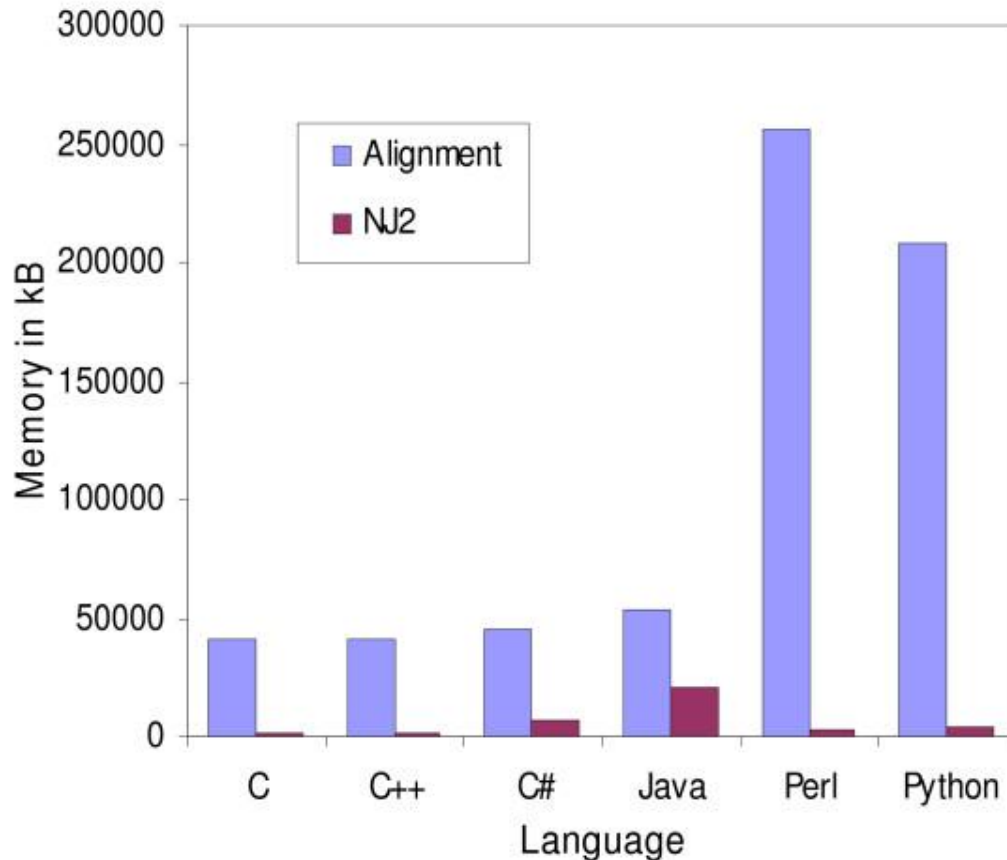
Speed comparison of the BLAST parsing program. Speed comparison of the BLAST parsing program implemented in C, C++, C#, Java, Perl and Python. The programs were run on Linux and Windows platforms. The input file was a 9.8 Gb file from a BLASTP run.

BMC Bioinformatics. 2008; 9: 82.

Memory usage comparison of the Neighbor-Joining and global alignment programs. Memory usage comparison for the Neighbor-Joining and global alignment programs implemented in C, C++, C#, Java, Perl and Python. The programs were run on a Linux platform.

M Number of lines for each program. Number of lines for the global alignment, BLAST parser and Neighbor-Joining programs implemented in C, C++, C#, Java, Perl and Python.

Course goals

- Understand C programming language in general
- Learn how to write good code
- Learn how to work together
- Practice, practice and practice…

Course contents

- Introduction to C programming language
- *Programming skills other than languages
  - Debugging tools
  - Using open source communities
  - Keeping projects documented and manageable
  - Source code managing
  - Parallel computing
  - Understanding the hardware
  - Value your time

*Dudley JT, Butte AJ (2009) A Quick Guide for Developing Effective Bioinformatics Programming Skills. *PLoS Comput Biol* 5(12): e1000589.

Course organization

- Course introduction (Week 1)
  - Code editor: Emacs
- Part I: Introduction to C programming language (Week 1 - 12)
  - Chapter 1: Overall Introduction (Week 1-4)
    - C
    - Unix/Linux
  - Chapter 2: Types, operators and expressions (Week 5)
  - Chapter 3: Control flow (Week 6)
  - Chapter 4: Functions and program structure (Week 7-8)
  - Chapter 5: Pointers and arrays (Week 9)
  - Chapter 6: Structures (Week 10)
  - Chapter 7: Input and Output (Week 11)
- Part II: Skills others than programming languages (Week 12- 14)
  - Debugging tools（Week 12-13）
  - Keeping projects documented and manageable （Week 14）
  - Source code managing （Week 14）
- Part III: Reports from the battle field (student forum) (Week 15 – 16)

Course organization (2)

- Thursday 14:00-15:40 (Every Week)
  - 生物药楼4号楼-225，生信实验室
- Tuesday 8:00-9:40 (Even weeks)
  - 生物药楼4号楼-225，生信实验室

Course features：
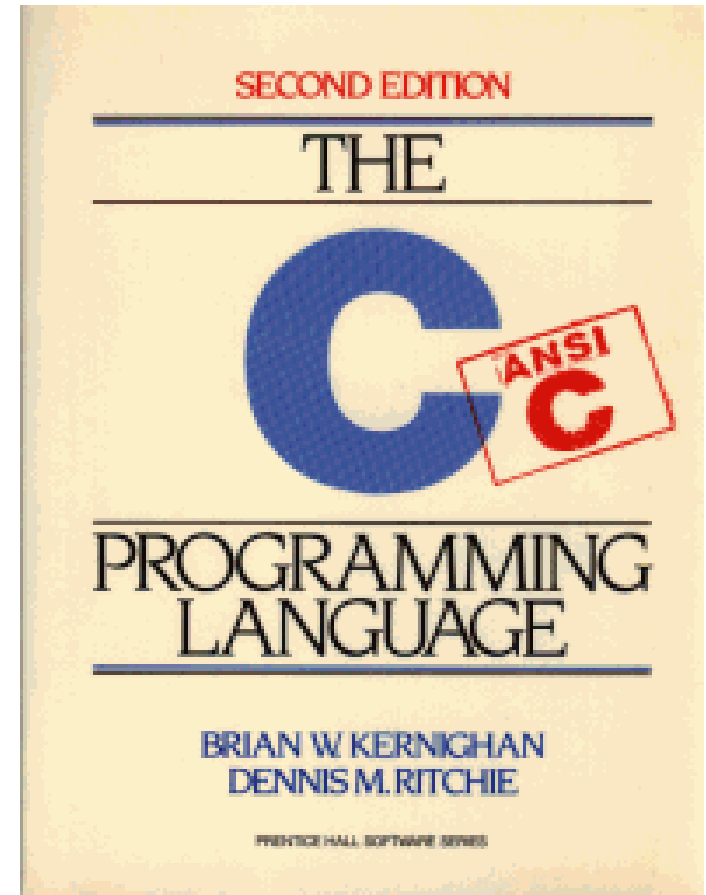# this is a practice course and you need lots and lots of practices

Prerequisites

- Mathematics (no)
- Computer Science (no)
- Biology (no)

Text Book

The C Programming Language, Second Edition by Brian W. Kernighan and Dennis M. Ritchie. Prentice Hall, Inc., 1988.

References

- Text Editors
  - Emacs
    - tutorial: http://www.gnu.org/software/emacs/tour/
    - Manual: http://www.gnu.org/software/emacs/manual/emacs.pdf

- Debug Tools
  - GDB
    - Document: http://www.gnu.org/software/gdb/documentation/

Grading

- Homework   50%
- Projects   30%
  - Design and implementation of a bioinformatics tool
- Presentation   20%

作业规定

- 作业允许合作，但是必须注明各人的贡献
- 作业报告必须用自己的语言独立完成
- 严禁抄袭
  - 抄袭者：不及格(F)
  - 被抄袭者：成绩降一级（A→B, B→C, C→D, D→F）

Course website

- http://cbb.sjtu.edu.cn/~ccwei//pub/courses/2014/plb/plb.php

- If you have any question, send me an email at: ccwei@sjtu.edu.cn