

General Requirements for Programming Assignments (Gerstein Section)

Students will have a choice of four languages in which to implement their programming assignments. These languages are listed below. Please note that the version number given after each language is the compiler/interpreter version that we will be used to test your code. If the program does not run under the listed compiler version, the programming assignment can not be graded and will receive 0 points.

Permitted languages:

- Perl (version 5.8.8)
- Python (version 2.6.2)
- C++ (gcc version 4.1.2)
- Java (version: 1.6.0)

A Note About Packages

The goal of this course is to become familiar with and implement a variety of bioinformatics algorithms. Hence, using a package that performs Smith-Waterman alignment in an assignment to implement Smith-Waterman (for example) is not allowed. Please get approval from the TAs **beforehand** for using any packages that are not part of the default language installations.

Submitting your programs for grading:

You are required to submit:

- 1) Final, commented source code
- 2) README file with instructions on how to run your program (this includes the exact command-line instructions to compile and run your program). These instructions should be complete, as the TAs will not guess.
- 3) Test run of your implementation. Pay special attention to the assignment-specific instructions regarding the output format.
- 4) Brief summary of how your program works (i.e. what it does and how it does it).

Assignments should be e-mailed in zip format to cbb752@gersteinlab.org.

Due Dates:

As programming assignments are given at least a week before the due date, the due date/time is **strict**. No extensions will be granted (i.e. if a program is due at 5 PM and we receive it at 6 PM, do not expect full credit).

Other notes:

Input files must be used as provided. Also, programs should not hard-code input and output file names. This is bad form.

Although there are no specific runtime requirements, programs that take hours to run will not be graded.

Yale's policies regarding academic dishonesty and plagiarism also apply.

This course assumes prior programming knowledge and hence is not an "Introduction to Programming" course.

For any **logistical** questions, please ask a TA.