1.1 Genetics & Computer Programming

1.1.1 Why should biologists learn to program?

With the constant advancement of technology used to analyze biological data, such as genomic sequencing and proteomics spectroscopy, biologists are generating more and more data. Instead of focusing on one gene at a time, for example, genetic researchers are studying the entire genome at once. This high-throughput analysis allows for the exploration of multiple genomic aspects at once, but it also requires dealing with large amounts of data. In fact, biological data are being generated faster than analysis can occur. In many cases, data sets are so large they cannot be opened in standard spreadsheet software and require high-performance computing infrastructure in order to be analyzed. These analyses can occur much quicker and in a more memory-efficient manner by typing out commands to manage text-based data instead of clicking on buttons in standard software programs.

Programming is the act typing commands to instruct a computer to perform tasks. Programming is a learned skill very much like other biological skills, such as performing RNA extractions and purifying proteins. Biologists that learn how to program can perform analyses on the data they produce through experiments. A basic understanding of programming also facilitates collaboration with computational associates; biologists who know how to program can discuss the output of their experiments have a better understanding of how it needs to be analyzed. In addition, biologists can visualize the results of their analyses with more fine-tuned approaches when they know how to program. Programming skills are widely applicable to many fields within biology and beyond and are in high demand by most research labs and biotechnology industries.

1.1.2 Computer Programming

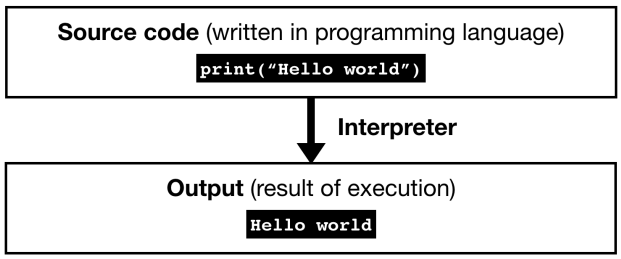
A **computer program** is a set of commands that provide instructions to a computer to perform a specific task. Every action you take on a computer requires a computer program to complete the task, including opening this PDF file. **Computer programming** is the act of designing and implementing a computer program.

Thousands of computer programs have been written to perform common tasks in biological research. This includes programs to map sequencing reads to genomes, programs to predict protein structures, programs to analyze microscopic images, and programs to perform common statistical analyses in genetic research.

Computer programs are written in **programming languages**. Instructions written in a programming language are called **code**. Each programming language has a specific set of rules that must be followed in order to write functional code. Like English and other natural languages, words and symbols in programming languages have meaning. This is called the **semantics** of a language. Specific words and punctuation marks have particular meanings in code that differ from their meanings in natural languages and that can differ

between programming languages. The order in which words and symbols are written is the **syntax** of a language. The structure and arrangement of words and symbols (syntax) along with their meaning (semantics) determines the function of a piece of code.

In order for a computer program to perform a task, the program must be **executed**. To execute code is to carry out the instructions described by the code. Executing code is also referred to often as "running code". The **interpreter** is a program that reads and executes code written in a programming language.



Code written in a programming language is called **source code**. When the code is executed by the interpreter, the task defined by the code is performed which leads to some form of **output**. Output is the result of executing code.

One of the many programming languages is **R**. R was designed for performing statistical analyses and creating graphics. In this course, the first seven weeks will focus on learning to write code in the R programming language. R is commonly used by biological researchers as it contains built-in methods for performing statistical tests and creating plots, allowing researchers to easily analyze and visualize many types of biological data. R is also designed to work with data in spreadsheet format (tables of rows and columns). Lots of biological data is stored in this type of format, for example, the rows of a table representing individual samples and columns representing measurements and features for each sample.

The functionality of R is enhanced by thousands of **packages**. An R package is a collection of functions and/or datasets that can be downloaded and installed to increase the utility of R. Many of the existing R packages were created specifically to aid in genomic analysis, including packages for analysis of RNA-sequencing data which will be taught in this course.

This course will also cover programming in the **command line**. A command line interface is an entirely text-based system for interacting with a computer. Many bioinformatic methods and tools are built to be run from the command line, including programs for sequence alignment, the processing of sequencing data, and working with files containing genomic features. Further details about the command line are in module 8.1.