

combined

June 7, 2020

1 Introduction

Welcome to the Microbial Genomics Lab for Fall 2020! This is a computer-based lab course which will teach you how to use computers to analyze microbial genomes and metagenomes.

1.1 Course Objectives

The overall goals of this course are to: (1) develop basic understanding of computational methods involved in analyzing microbial genomes, (2) become familiar with common bioinformatics tools used to (assemble, annotate, analyze) microbial genomes and metagenomes, (3) learn basic techniques to construct phylogenetic trees, (4) learn how to visualize microbial genomic data using Python and/or R, (5) learn basic Python programming.

Specifically, at the end of this course, you should be able to:

- use command line environment
- use Unix/Linux commands
- process raw NGS data into usable forms
- assemble microbial genomes
- assemble metagenomic data
- annotate microbial genomes
- construct phylogenetic trees to assess relationships between different microbes
- identify any given gene(s) of interest and use it for further downstream processes
- learn to write simple Bash, Python, and R scripts
- use Jupyter Lab to document computer exercises
- use git for version control

1.2 Materials

1.2.1 Required Text

None. Instructional materials and required reading will be provided through Blackboard or through a Wiki page. Computational exercises will be provided through online Wiki documentation system and Jupyter Lab notebooks.

1.2.2 Required Materials

A laptop computer. A user account on GW's Colonial One high-performance computing clusters. You need to apply for a user account through this website: <https://colonialone.gwu.edu/getting-access/>.

1.2.3 Additional materials

Additional reading materials and media will be provided on Blackboard. These may include research journal articles, news articles, YouTube videos, etc.

1.3 Assessment of Learning

1.3.1 Attendance/In-class Participation

Attendance will be taken and will count towards the final grade.

1.3.2 Assignments

In addition to in-class computer exercises, students will have computer assignments to complete before the next class period. There may also be assigned readings and materials from the assigned readings may be on the midterm or the final examinations.

1.3.3 Documentation using Jupyter Lab

We will make extensive use of Jupyter Lab to document computer exercises and also to keep track of class progress. This Jupyter electronic notebook will need to be turned in at the very end of the semester and will be graded and the scores will count towards the final grade.

1.3.4 Final Project

Students will need to pick a project assigned by the instructor or their own choosing (if they have their own data to analyze, for example). Findings from the project will need to be typed in the form of a research manuscript. This project will count towards the final grade.

1.3.5 Final Presentation

Students will present their individual (or group) final projects in the form of a Powerpoint presentation.

1.3.6 Exams

There will be two exams: a midterm and a final. These exams will test students' comprehension of lectures and computer exercises given. The exams may consist of questions to be answered through Blackboard and computational exercises to be answered using a Jupyter notebook.

[]:

2 Week 1

This is an introduction to BISC 4236 (Microbial Genomics Lab). Testing Jupyter lab capabilities.

2.1 Setting up environments for Mac Users

2.1.1 Install Xcode

Here, you should first install Homebrew on Mac.

1. Open your terminal
2. Type this: `xcode-select --install`
3. If this doesn't work, follow the instructions from this page:
(<https://osxdaily.com/2014/02/12/install-command-line-tools-mac-os-x/>)

2.1.2 Install Homebrew

- Go here and follow the instructions: (<https://docs.brew.sh/Installation>)
- Open your terminal
- Type:

```
cd
```

```
mkdir homebrew && curl -L https://github.com/Homebrew/brew/tarball/master | tar xz --strip 1 -&
brew install coreutils findutils gnu-tar gnu-sed gawk gnutls gnu-indent gnu-getopt grep
```

2.2 Setting up environments for Windows Users

Try installing Windows terminal:

<https://devblogs.microsoft.com/commandline/windows-terminal-1-0/>

<https://www.microsoft.com/en-us/p/windows-terminal/9n0dx20hk701?rtc=1&activetab=pivot:overviewtab>

Hopefully, this will be it for command line tools installation. One thing to remember, Unix/Linux commands are case-sensitive. All of them should be typed in lower-case in general unless otherwise instructed.

2.3 Install Miniconda and Bioconda

Go here to download Miniconda package manager: <https://docs.conda.io/en/latest/miniconda.html>

Follow instructions for both Mac and Windows. Choose Python 3.7 version. For Mac, I recommend you download the “pkg” version which is a graphical version of the installer.

Close your terminal and open it again. Try typing conda to see if the command works. You will see a bunch of options you can type with the conda command.

Install bioconda by typing the following commands in your terminal:

```
conda config --add channels defaults
conda config --add channels bioconda
conda config --add channels conda-forge
```

These commands will add bioinformatics related tools “channels” to miniconda. Now, you should be able to search and install tools such as “BLAST” on your computer by just typing:

```
conda search blast
conda install blast
```

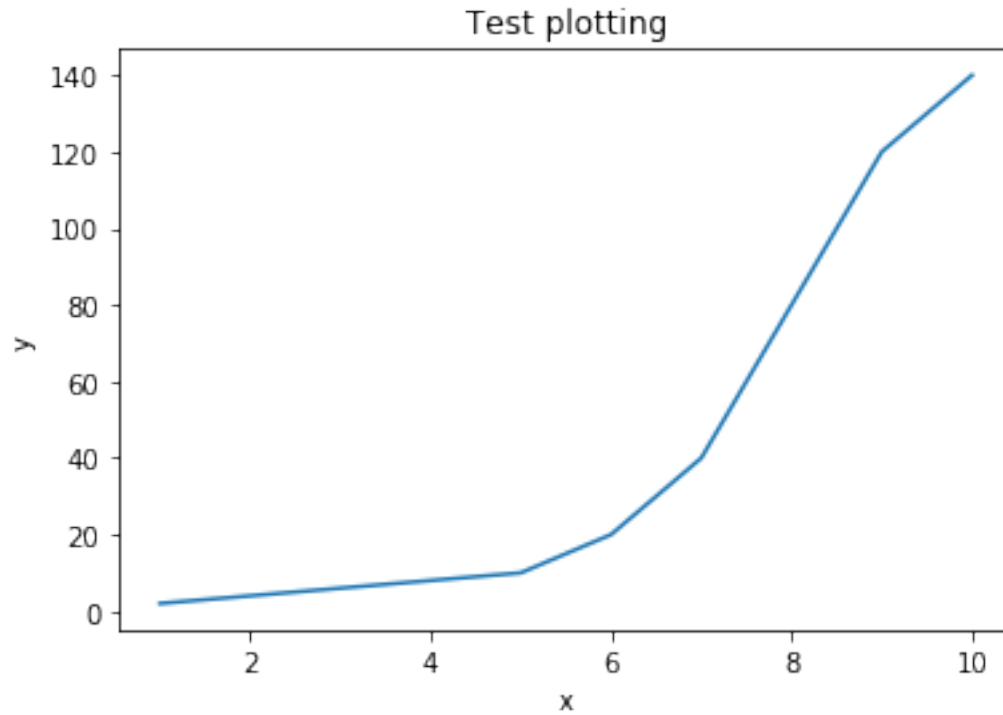
2.4 Test Python code

```
[2]: import matplotlib.pyplot as plt
import matplotlib.pyplot as mpl

x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
y = [2, 4, 6, 8, 10, 20, 40, 80, 120, 140]

plt.title("Test plotting")
mpl.xlabel("x")
mpl.ylabel("y")
plt.plot(x, y)
```

```
[2]: [<matplotlib.lines.Line2D at 0x118561850>]
```



[]:

3 Week 2

3.1 Using Colonial One

This page shows how you can use Colonial One to run things.

3.2 Obtaining accounts

You go here to request an account:

<https://colonialone.gwu.edu/>

3.3 An example job submission script

```
#!/bin/sh
# one hour timelimit:
#SBATCH --time 1:00:00
# default queue, 32 processors (two nodes worth)
#SBATCH -p defq -n 32
```

```
module load openmpi
```

```
mpirun ./test
```

```
[ ]:
```