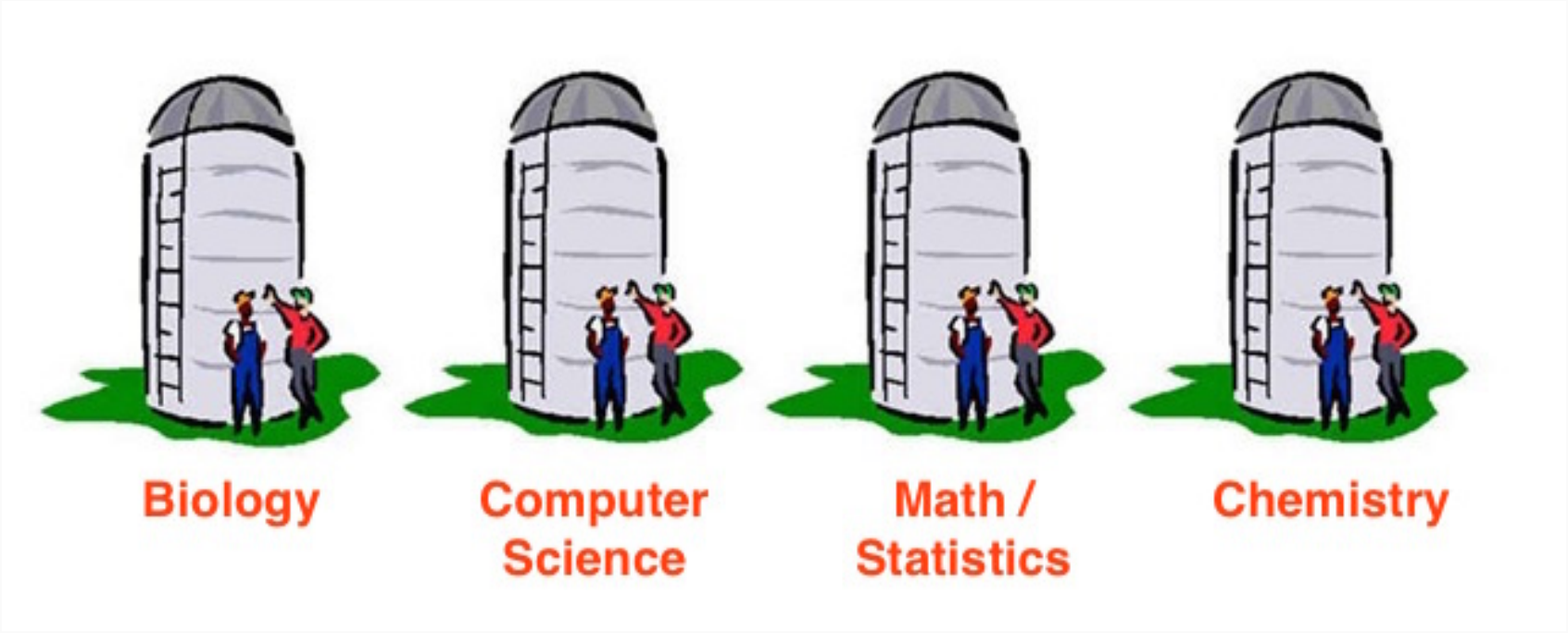
COMP 383 Computational Biology

Introductory Lecture

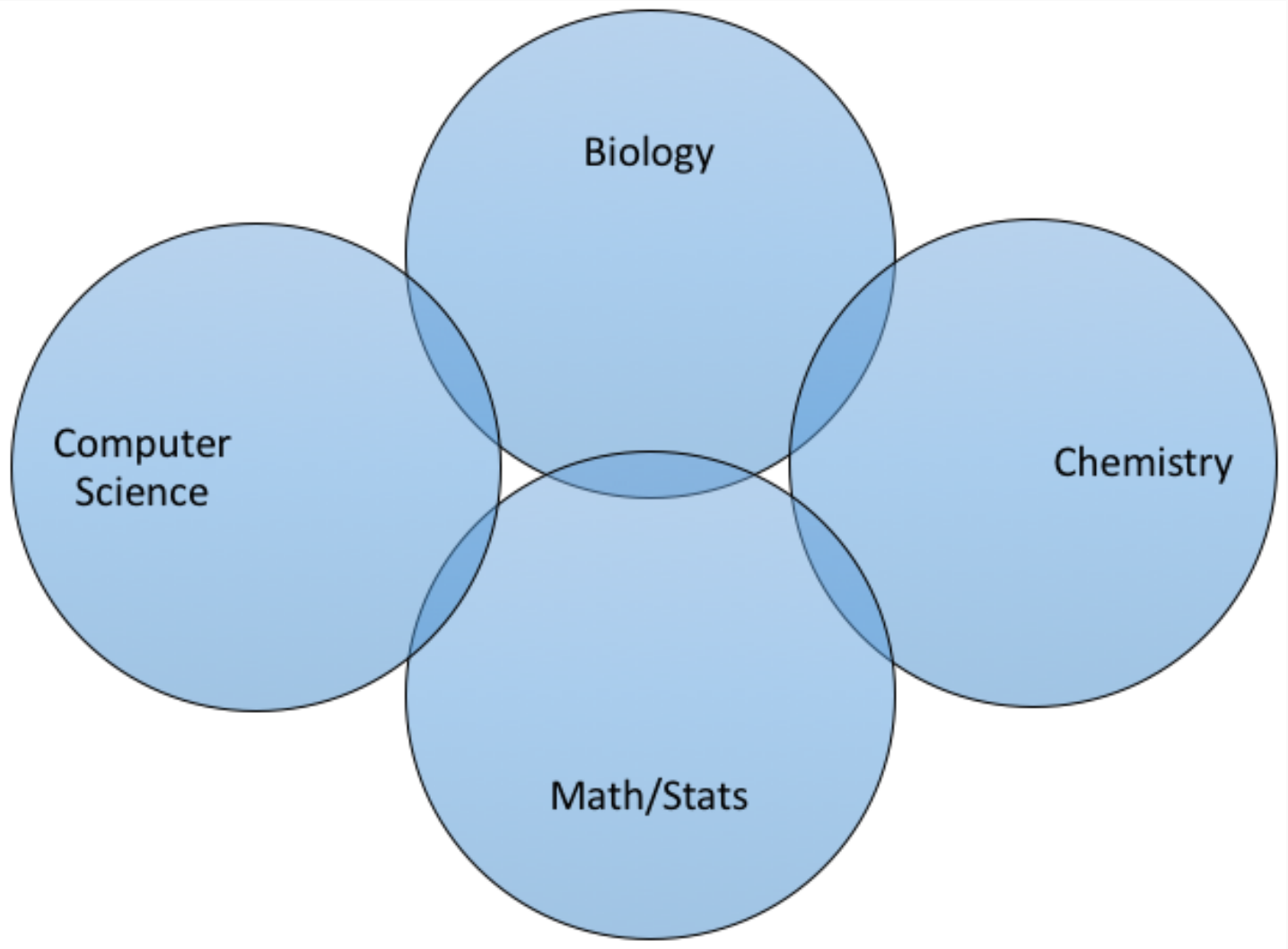
Spring 2016

Loyola University Chicago
hwheeler1@luc.edu

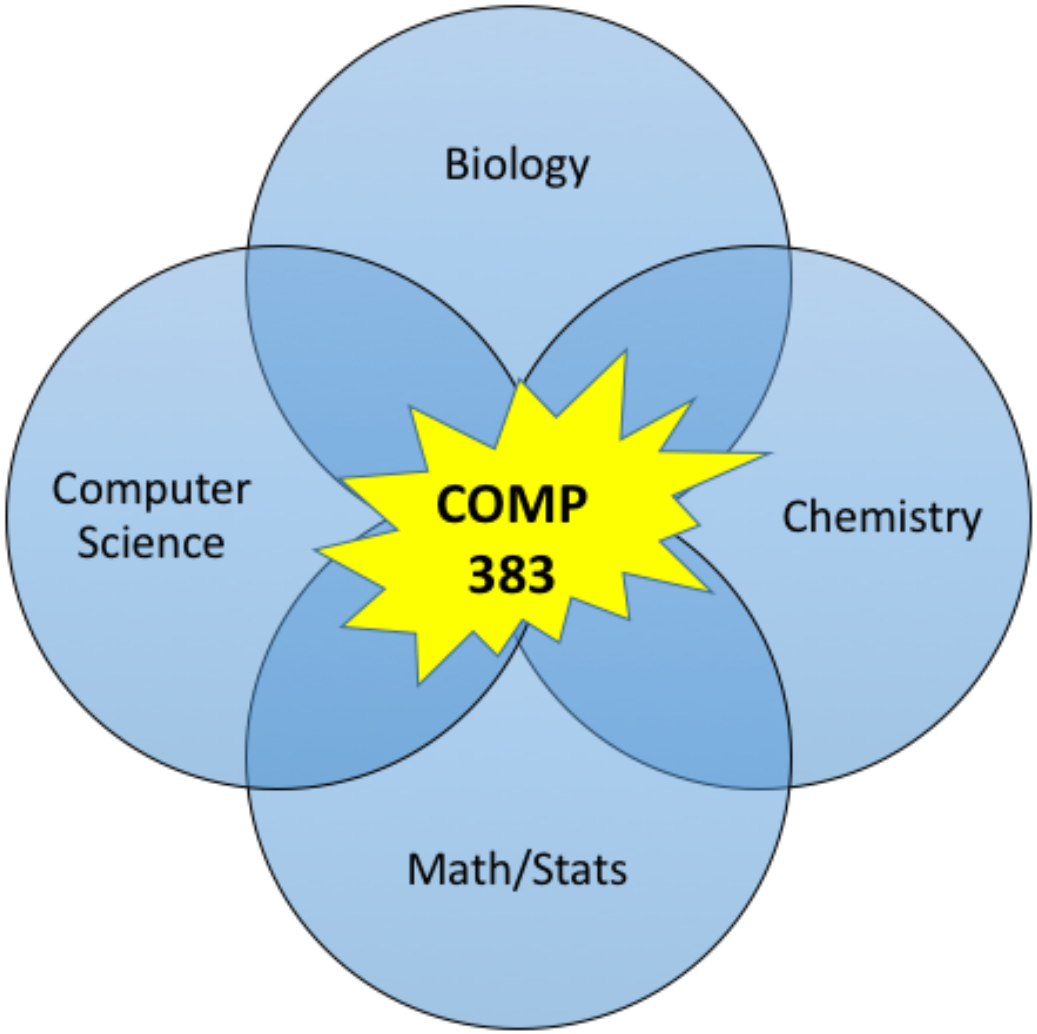
Bioinformatics Major Courses



In BIOL 388 Bioinformatics, you started to see an intersection of these fields



COMP 383 Computational Biology integrates all you've learned

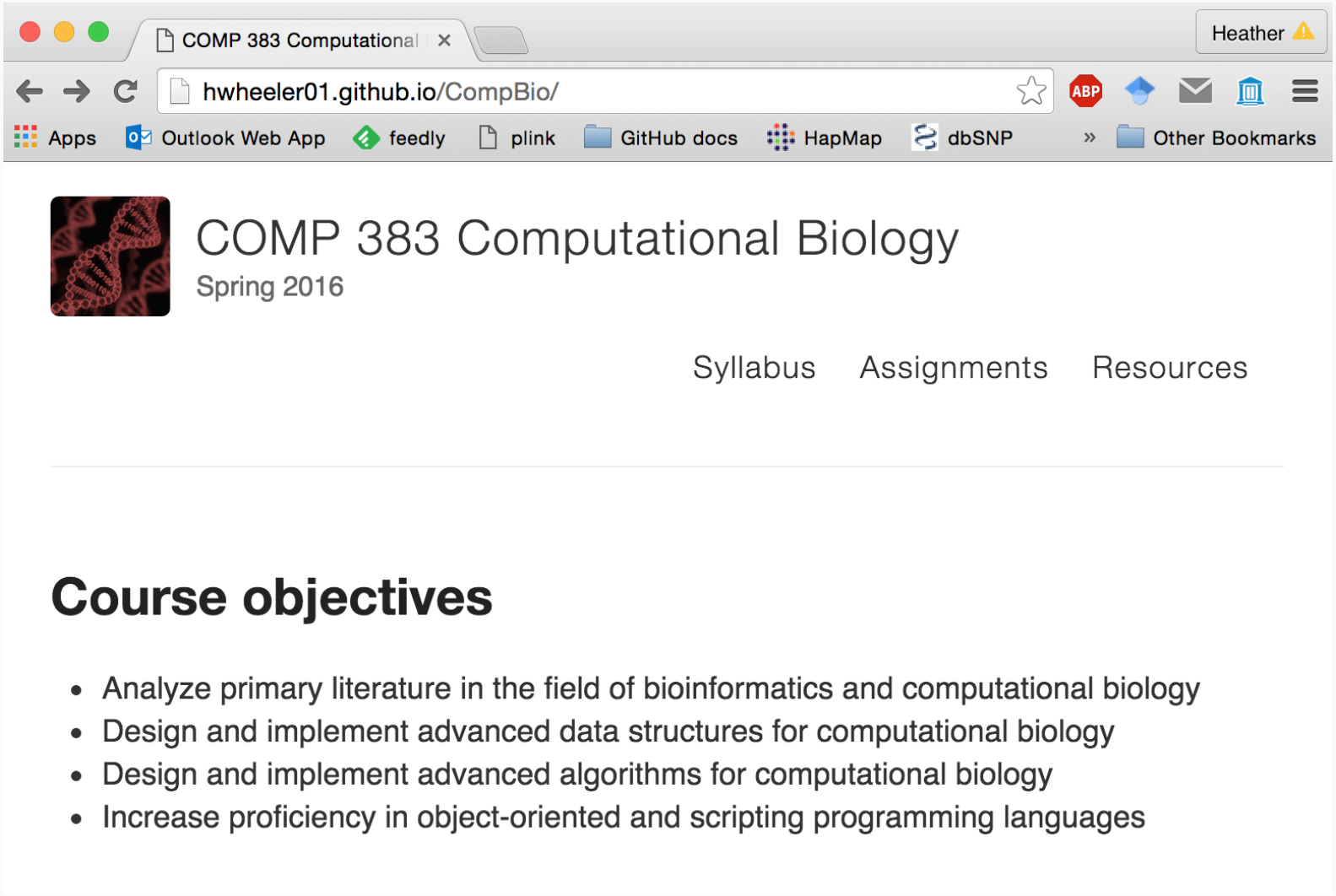


COMP 383 Course Structure

1. First few weeks, learn Python, apply it to common biological problems (toy examples) - 20%
2. Group ***Research*** Project: You will be given freedom to work on a computational biology problem - 80%

- This could involve automating a research pipeline or building an analysis tool
- This is a “real life” research project (options from Loyola faculty will be available), so data will be messy and unanticipated obstacles will occur
- It will be important to set benchmarks and stay on task in class
- Don’t procrastinate, get things done and strive to make your code better
- Learn all you can, the future of biology/medicine/data analysis needs you!

The syllabus and assigments are posted on the class website



http://hwheeler01.github.io/CompBio/

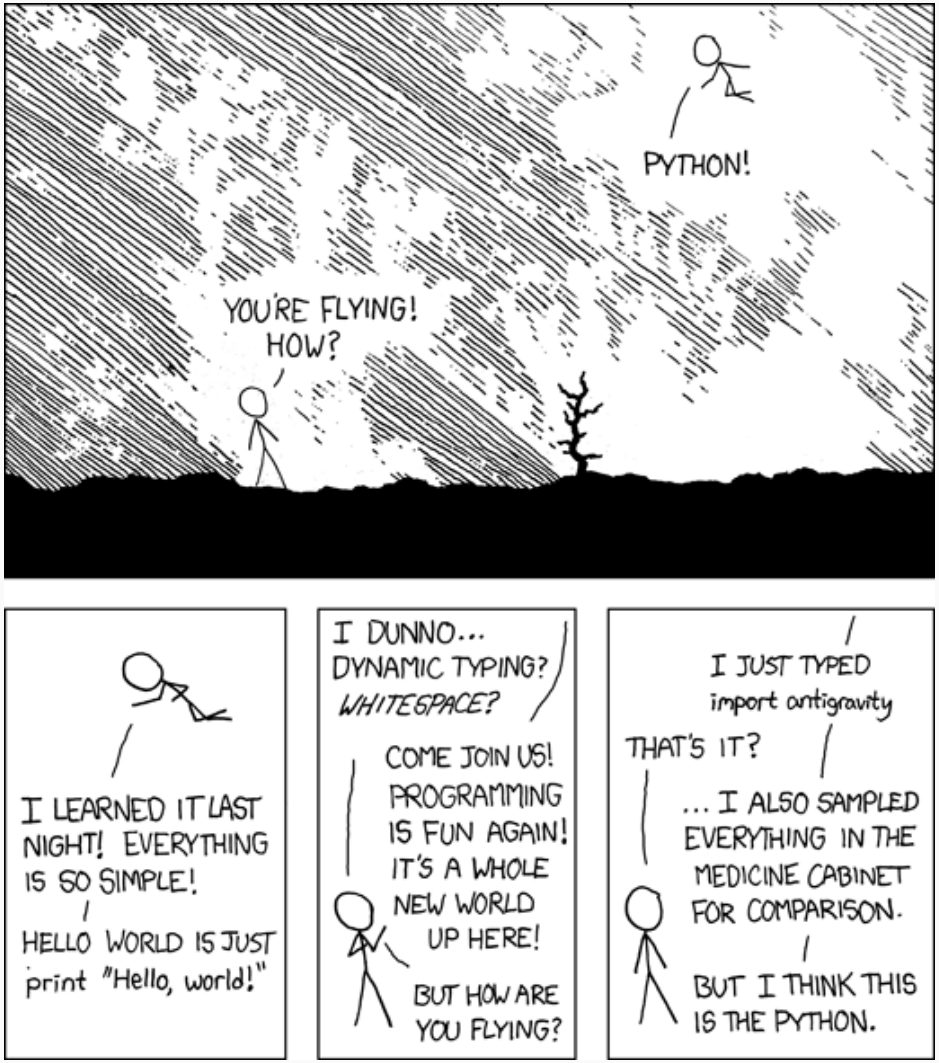
Why Python?

1. Everybody’s doing it. It has a large (and growing) user base among scientists.
2. Python is a general purpose, high level, programming language – you should be able to do anything you want to do using Python, and it should be relatively easy to accomplish.
3. It’s free, well-documented, and runs on all operating systems.

Why Python?

4. It’s easier for novices to pick up than most other languages and it’s also used by many professional programmers. This makes collaborating with both novices and experts easier.
5. A dynamic language like Python allows us to write small programs quickly and to also manage the complexity of larger ones.
6. If we want to squeeze every last ounce of performance out of our hardware, then a compiled language like C++ is better, but if we want to quickly answer a research question, which is usually the case in computational biology, Python is often easier/faster.

Why Python?



http://xkcd.com/353/

Python 2.7

For this class, I recommend Python version 2.7 rather than Python 3.x

- ROSALIND assumes 2.7
- BioPython for Python 3.x is not as stable, but improving all the time
- If you’re used to Python 3, feel free to use it, just note some of the ROSALIND hints may require slightly different syntax

Python 2.7 Strings

```
print "Hello, World!"
a = "Hello"
b = "World"
type(a)
print a + ", " + b + "!"*3
a[0:4]

## Hello, World!
## <type 'str'>
## Hello, World!!!
## 'Hell'
```

Python 2.7 Numbers

```
a = 12
b = 2.5
c = a + b
type(a)
type(b)
print str(a) + " + " + str(b) + " = " + str(c)

## <type 'int'>
## <type 'float'>
## 12 + 2.5 = 14.5
```

Python 2.7 Division

```
18/5
18/5.0
18/float(5)
```

```
## 3
## 3.6
## 3.6
```

Or place this import at the beginning of the file:

```
from __future__ import division
18/5
```

```
## 3.6
```

Start first assignment in ROSALIND

http://hwheeler01.github.io/CompBio/assignments/