Rationale for this course

- Life sciences are becoming a data driven science
- Data is represented as text files in various formats that are transformed one step at a time
- Most bioinformatics classes are focused on computer science or algorithms.
- We will focus on information processing and applications

Course Requirements

- Recommended: newest Mac OSX Maverick 10.10.5
- Or a Linux based operating system
- Windows will NOT work.
- You may be able to install Linux on your Windows machine.
- This is a good time to tell your advisor to buy you a Mac ... a nice Macbook Air that will make you superbly productive
Lecture topics

15 weeks – two lectures per week = 30 lectures

- core informatics competency
- computational foundations
- biological data formats
- elements of statistical methods and visualization
- software tools and their applications

Lecture Formats

- Each week we try to cover one topic
- Background information
- Practical examples that tie in with the topic
- Finishing with in class exercises + homework
- We’ll make it simple and easy to follow

Homework determines your grade

- Home work will be given out during each lecture and correspond to the lecture. Will be labeled as 1, 2 ... 30
- Home work is due on the Tuesday of the following week of when it was given out.
- For example: homework 1 and 2 will be due next Tuesday. Submit homework as printouts. Staple both home work together and label with your name.
- Note: there are two office hours between each homework’ due date (Wed and Mon)
- Homework usually fits on one sheet of paper. Show the commands and their output. You can print screens but don’t print white text on black background.

Grading

- Grades will be the average of all homework
- For homework you may work in teams – but everyone needs to perform the actions themselves.
- Don’t just copy paste someone else’s code.
Complexity versus decision making

- Most bioinformatics analyses consists of a very large number of simple decisions
- Most of which need to be correct!
- This is what makes it difficult
- There are no strict rules, only guidelines → you must be able to improvise and adapt

Expectations

You can only learn by doing it

Spend 3-6 hours outside class each week:
- Explore behaviors
- Expand the scope of the study
- Try new solutions

Time flies when you know what you are doing.

BioStar : http://www.biostars.org

- I started the site in 2009 during the first year that BMMB 597D was offered!
- It was meant to support questions for this course
- Today it has grown to attract over 180K unique visitors per month and over 10 million page views per year
BioStar : http://www.biostars.org

Onto genome sequencing

- What is a genome?
- Why do we want to study it?
- What can we explain via a genome?
- What complications might be there?

I’ve read that the Genome is the Book of Life?

Do you think that is true?

YES      NO

Bioinformatics today

Large datasets generated by complex equipment

1. Data management → storage, transfer, data transformation → domain of Information Technology
2. Data analysis → mapping, assembly → algorithm scaling → domain of Computer Science
3. Statistical challenges → traditional statistics is not well suited for modeling systematic errors over large number of observations → domain of Statistics
4. Biological hypothesis testing → data interpretation domain of Life Science
How is bioinformatics pursued?

<table>
<thead>
<tr>
<th>Command Line Tools</th>
<th>R Programming Environment</th>
<th>Tools with graphical User Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Action words” are chained together to form a pipeline. Data &quot;flows&quot; from one command to the other.</td>
<td>A high level programming statistical environment. Attempts to provide with “simple” constructs that perform complex tasks. Excellent visualization capabilities.</td>
<td>A Discoverable Environment that &quot;appears&quot; to be simple. Often lag far behind other options. Difficult to understand data provenance – what has been done to a given dataset.</td>
</tr>
<tr>
<td>Resembles natural language. Provides generic building blocks. Very adaptive and expressive.</td>
<td>Driving principle: simplicity and specialization, do one thing but do it right.</td>
<td>Quite difficult to repeat the same process multiple times.</td>
</tr>
<tr>
<td>Requires a deeper understanding of how computers work behind the science.</td>
<td>Driving principle: encapsulate complex, multistep tasks into a single command. One magic incantation: full analysis done.</td>
<td>“White lies” and exaggerations are common</td>
</tr>
</tbody>
</table>

How to turn your computer (and of course yourself) into a computational beast

**On a Mac you will need to:**

1. Update your **Mac OS** to the latest version 10.10.5 (Yosemite)
2. Using the App Store download and install **XCode**
3. You will need to install the **command line tools** (see next slide)

**On Linux**

- Install a well supported linux version: **Ubuntu**, **Debian**, **Fedora** etc.

- Use a **package manager** (**apt-get, yum**) to install dependencies like so (Ubuntu):

  `apt-get install zlib1g-dev`

- See the course webpage for the full command line to
Success looks like this

Open a terminal: the `make` tool and `gcc` both work.

Directory Structure

- You must develop a firm understanding of how files are organized on a computer.

  Command line concepts:
  - The root is represented by the `/` symbol.
  - A "tree" like structure where branches are separated by the `/` symbol.
  - Going "back", "or left" (up) is the `../` symbol.
  - The location where you are (current directory) is represented by the `. ` symbol.

```
/a/b/c
/a/b/../e/f
./g/h/l
./b/./e/f
```

Organizing your projects

The UNIX command line

Action words

Chain words together to form statements

Open Terminal on Mac.
Getting to know the terminal: man (manual)

Discover-ability: learn how to find out more details on a tool process

Learn more about these commands

- `ls` (list directory contents)
- `rm` (remove files/directories)
- `cp` (copy files)
- `cd` (change directory)
- `mkdir` (make directory)
- `rmdir` (remove directory)
- `pwd` (print current work directory)

Note that each one of these commands has options (flags)

Customizing commands

- A flag is a small decoration used to change or customize what a tool does.

Summary

- Understand course requirements
- Set up your computer
- Get started with simple terminal based commands
Homework 1

1. Install all the required software for your computer.
2. Show that you can navigate and list the contents of directories
3. Create and delete directories, create and delete files
4. Find help on commands.
5. Understand what flags to commands do.
6. What flags will make the *ls* command write out the files sizes in “human friendly” mode?
7. What flag will make the *rm* command ask for permission when removing a file?
8. What flag will make the *cp* command ask for permission if the copy would overwrite an existing file (this is called clobbering)?
9. Make the directories *src* and *bin* and *refs* from your home directory. These will store programs and analysis results.