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 OS X Maverick 10.9.4
- Or any other Unix based operating system
- If you have a Windows computer please install Linux
  - Ubuntu on a USB stick, Ubuntu Live CD
  - Use VirtualBox and install Linux into it
  - Or set up dual boot Linux and Windows

Lecture topics

15 weeks – two lectures per week = 30 lectures

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

Lecture Formats

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

Homework

- 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.
- 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 if you want to but:

  *please don’t print a console output of white text on a black background*
Grading

- Grades will be the weighted average of all homework + final project

- The final project will (weight of 20%) will similar to but more complex than a typical homework will make use of concepts learned thought the year

- Final project given out last week Thursday, and will be by the Monday on the final’s week.

- For homework and final project you may work in teams – but watch out for “group think”.

Computation ↔ Thought

- Computational approaches reflect and affect the thought process

- When we learn informatics, we learn how to think in a way that is easy to translate into computation

- There is no magic – it is just like any other subject matter – it needs a lot of practice (the brain is a muscle)

- Similar to learning a foreign language – there is a vocabulary, a grammar → idiomatic expressions

Realistic News

Bioinformatics will never be easy or trivial!

It is like high altitude mountain hiking. Even if you have done it before does not mean next time is trivially easy.

There are only so many mountains you can climb in a week/month/year.

It always requires attention and patience and attention. Many things do go unexpectedly all the time.

WHY?

It is about information

- “Life is complicated” there are few rules (if any)

- People/scientists have a “blind spots”, they focus on what they want to achieve and hope that everything else is not relevant

- Living organisms want to survive at all costs – it is none of their concern is to make it “humanly understandable”
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.

Complexity versus decision making

• Most bioinformatics analyses consists of a very large number of very 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

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 done?

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

- Random errors and systematic errors accumulate and compound during each step: sample extraction, library preparation, sequencing protocols and data analysis
- Large number of measurements make unlikely events common. Systematic errors interfere with statistical assumptions

Challenges: shifting terminology

What is the difference between a SNP (single nucleotide polymorphism) and a SNV (single nucleotide variation)?

A SNV is a private mutation while a SNP is a mutation that is shared amongst a population

At what point does a SNP turn into SNV?

Data produced by equipment

Example: 74 new SNVs (single nucleotide variations) per individual per generation

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 100K unique visitors per month and over 4.5 million page views per year
BioStar : http://www.biostars.org

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

On a Mac you will need to:

1. Update your Mac OS to the latest version 10.9.4 (Mavericks)
2. Using the App Store download and install XCode
3. Download and install the command line tools

On Linux

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

• Use a package manager to install dependencies. Leads to incantations such as. See the course webpage for the full command line

    apt-get install zlib1g-dev
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:
- A "tree" like structure where branches are separated by the `/` symbol
- Going "back" (up) is the `../` symbol
- The location where you are is the `/` symbol.

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

Organizing your projects

```
http://www.ploscompbiol.org/article/info:doi/10.1371/journal.pcbi.1000424
```

The UNIX command line

**Action words**

Chain words together to form statements

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

Discoverability: 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.

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.