

Biology, Computers & Python

Michael Schatz

Sept 3, 2013

QB Bootcamp Lecture I



Outline

Part 1: Overview & Fundamentals

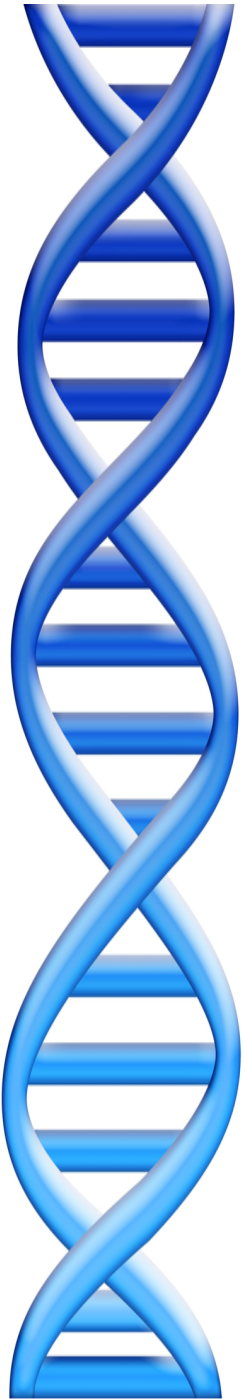
- Overview of Computer Systems
- Python Primer

Part 2: Sequence Analysis Theory

Part 3: Genomics Resources

Part 4: Unix Primer

Part 5: Example Analysis



Modern Biology Challenges



The foundations of biology will continue to be *observation, experimentation, and interpretation*

- Technology will continue to push the frontier
- Measurements will be made *digitally* over large populations, at extremely high resolution, and for diverse applications

Rise in Quantitative and Computational Demands

1. *Experimental design*: selection, collection & metadata
2. *Observation*: measurement, storage, transfer, computation
3. *Integration*: multiple samples, assays, analyses
4. *Discovery*: visualizing, interpreting, modeling

Ultimately limited by the human capacity to execute extremely complex experiments and interpret results

How do we draw conclusions?

- Comparison & Correlations: How does X compare to Y?

X	Y
Exomes of kids with autism	Exomes of kids that do not
Genomes of Europeans	Genomes of non-Europeans, mammals, ...
Gene expression in mutants	Gene expression in wild type
Firing patterns of mutant fly neurons	Firing patterns of wild type

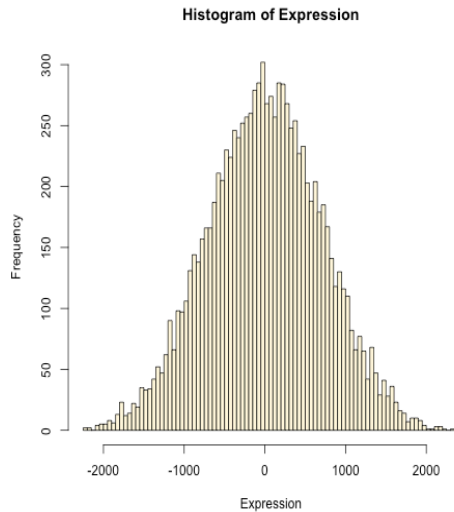
- Modeling & Predictions: How will X respond to Y?

X	Y
Mutant tomatoes	Increased temperatures
Human Microbiome	Probiotic treatments
Gene expression in mice	Knockout of transcription factor
Firing rate in flies	Decreased sodium levels

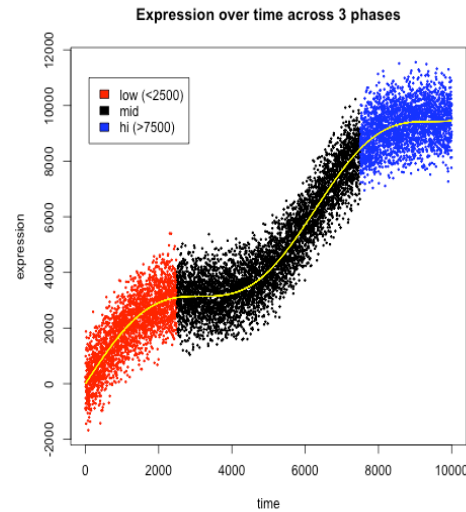
How do we DRAW conclusions?

-902.473
242.817
-872.453
73.9297
236.169
46.7525
975.014
716.563
-533.971
-120.282
725.12
-736.76
176.156
189.224
1847.46
-159.099
-56.4754
-973.626
1181.9
-315.455
-1480.43
215.293
-747.505
682.577
...

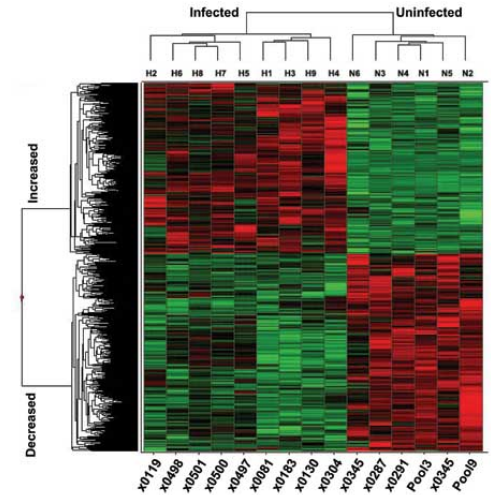
Histogram



Scatterplot



Heatmap

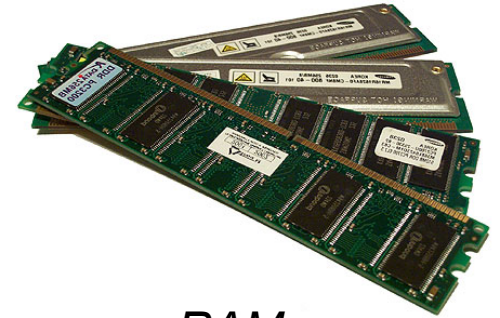


Data and data transformations are ubiquitous in science
Data are too numerous and transformations are too complex to do by hand
==> Mendel: 100k observations, 10 years
==> HiSeq 2000: 600B observations, 10 days
==> Make friends with your computational tools

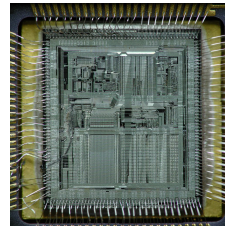
What is a computer? [hardware]



Hard Drive
Permanent Storage – 1TB
(big, slow, cheap)



RAM
Working Storage – 8 GB
(small, fast, expensive)



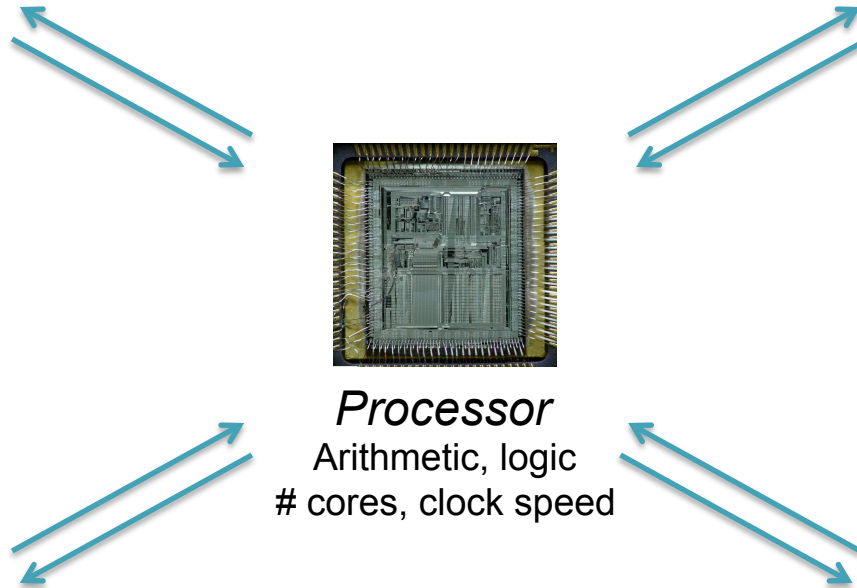
Processor
Arithmetic, logic
cores, clock speed



Display
Human Interface



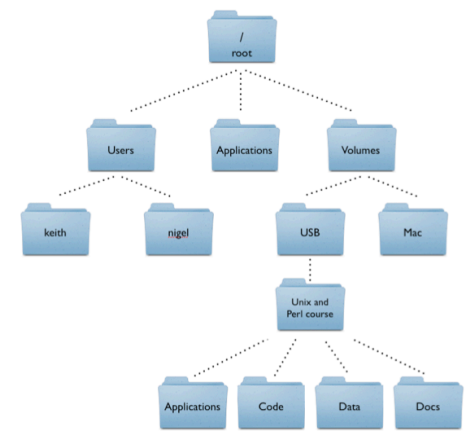
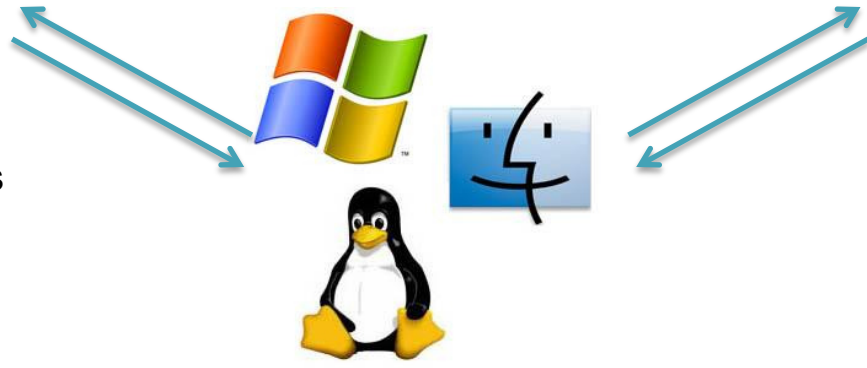
Network
Computer Interface
Home: 10Mb/s, CSHL: 1Gb/s



What is a computer? [software]

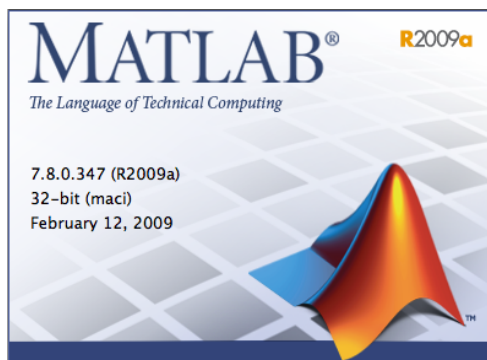


Office Applications
Presentations, Documents
Simple statistics and plots



Files / Data
Papers, sequences,
measurements

Operating System
Mission Control
Windows, Mac, Unix, iOS



Scientific Applications
Specialized Analysis
Commercial

```
iterations.m
Objective-C
Language Run Stop Console
18 time_t time1 = clock(); // Measure time from here
19 for (int i = 0; i < 100; i++) { // Do the test 100 times
20     NSArray *enumerator = [array objectEnumerator];
21     NSString *str;
22     while ((str = [enumerator nextObject]) != nil) {
23     }
24 }
25 time_t time2 = clock(); // Measure time from here
26 for (int i = 0; i < 100; i++) { // Do the test 100 times
27     for (NSString *str in array) {
28     }
29 }
30 time_t time3 = clock(); // Measure time from here
31 for (int i = 0; i < 100; i++) { // Do the test 100 times
32     for (int i = 0; i < [array count]; i++) {
33         NSString *str = [array objectAtIndex:i];
34     }
35 }
36 time_t time4 = clock(); // Measure time from here
37 double t1 = (((double)(time2-time1))/CLOCKS_PER_SEC)*1000;
38 double t2 = (((double)(time3-time2))/CLOCKS_PER_SEC)*1000;
39 double t3 = (((double)(time4-time3))/CLOCKS_PER_SEC)*1000;
40 printf("NSEnumerator: %0.1f ms\n", t1);
41 printf("Fast enumeration: %0.1f ms\n", t2);
42 printf("For-loop: %0.1f ms", t3);
43 return 0;
```

Code / Scripts
Research Applications
Academic

Programming 101

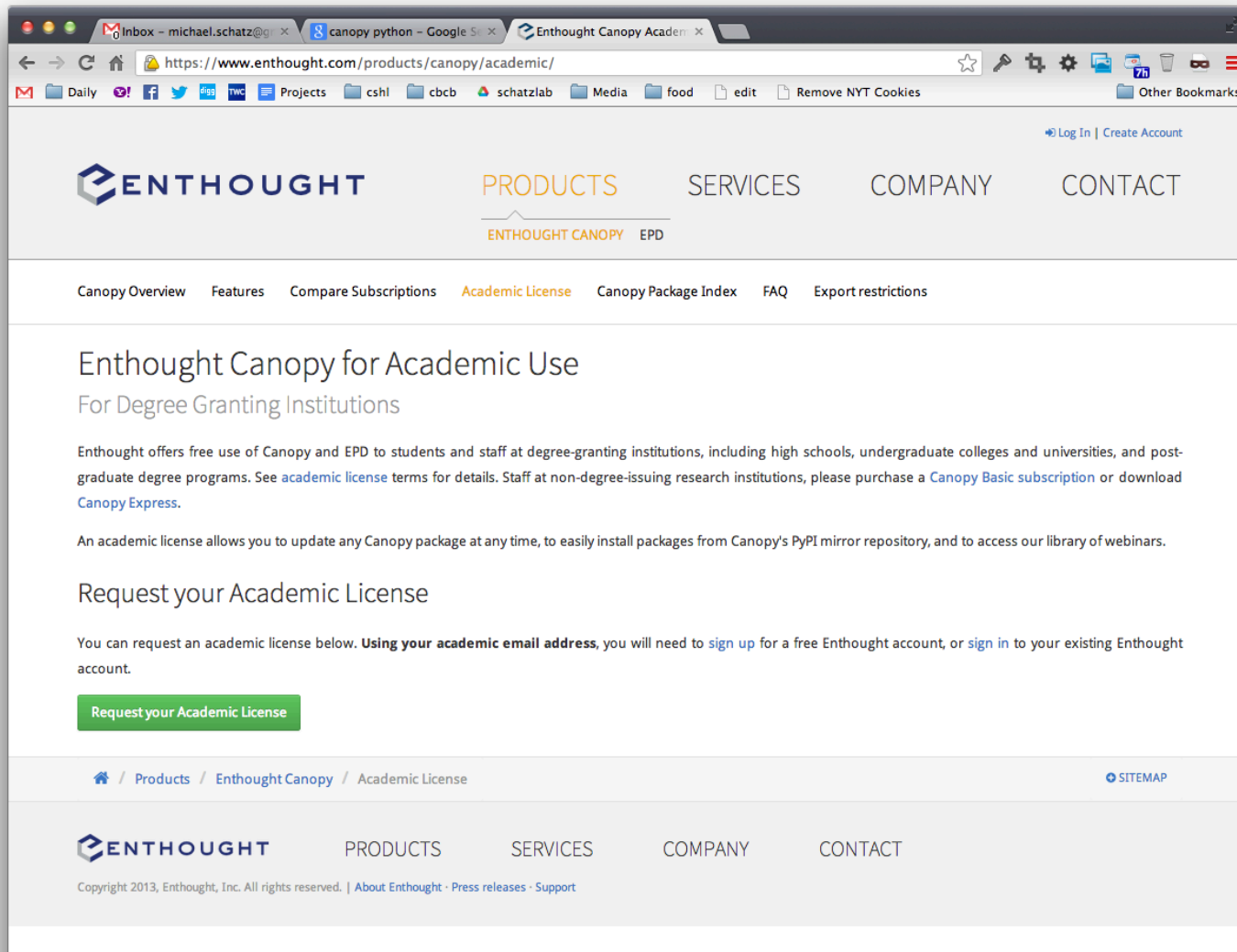
Mozart
Sinfonia Concertante in Eb
for Violin and Viola
K. 364
Allegro maestoso.

www.viola-in-music.com

```
iterations.m
Objective-C
Language Run Stop Console
18 time_t time1 = clock(); // Measure time from here
19 for (int i = 0; i < 100; i++) { // Do the test 100 times
20     NSArray *enumerator = [array objectEnumerator];
21     NSString *str;
22     while ((str = [enumerator nextObject])) {
23     }
24 }
25 time_t time2 = clock(); // Measure time from here
26 for (int i = 0; i < 100; i++) { // Do the test 100 times
27     for (NSString *str in array) {
28     }
29 }
30 time_t time3 = clock(); // Measure time from here
31 for (int i = 0; i < 100; i++) { // Do the test 100 times
32     for (int i = 0; i < [array count]; i++) {
33         NSString *str = [array objectAtIndex:i];
34     }
35 }
36 time_t time4 = clock(); // Measure time from here
37
38 double t1 = (((double)(time2-time1))/CLOCKS_PER_SEC)*1000;
39 double t2 = (((double)(time3-time2))/CLOCKS_PER_SEC)*1000;
40 double t3 = (((double)(time4-time3))/CLOCKS_PER_SEC)*1000;
41 printf("NSArray: %0.1f ms\n", t1);
42 printf("Fast enumeration: %0.1f ms\n", t2);
43 printf("For-loop: %0.1f ms", t3);
44
45 return 0;
```

A software program is like sheet music for the orchestra inside your computer
Static, written representations of an active process

Programming with Python



The screenshot shows a web browser window displaying the Enthought Canopy Academic Use page. The browser's address bar shows the URL <https://www.enthought.com/products/canopy/academic/>. The page features a navigation menu with links for Log In, Create Account, PRODUCTS, SERVICES, COMPANY, and CONTACT. The main content area is titled "Enthought Canopy for Academic Use" and includes a sub-heading "For Degree Granting Institutions". The text explains that Enthought offers free use of Canopy and EPD to students and staff at degree-granting institutions, including high schools, undergraduate colleges, and universities. It also mentions that staff at non-degree-issuing research institutions should purchase a Canopy Basic subscription or download Canopy Express. A section titled "Request your Academic License" provides instructions on how to request a license, including the requirement to use an academic email address and to sign up for a free Enthought account or sign in to an existing one. A green button labeled "Request your Academic License" is visible. The footer of the page includes the Enthought logo, navigation links, and copyright information: "Copyright 2013, Enthought, Inc. All rights reserved. | About Enthought · Press releases · Support".

<https://www.enthought.com/products/canopy/academic/>
<http://www.codecademy.com/tracks/python>

Questions?

<http://schatzlab.cshl.edu>