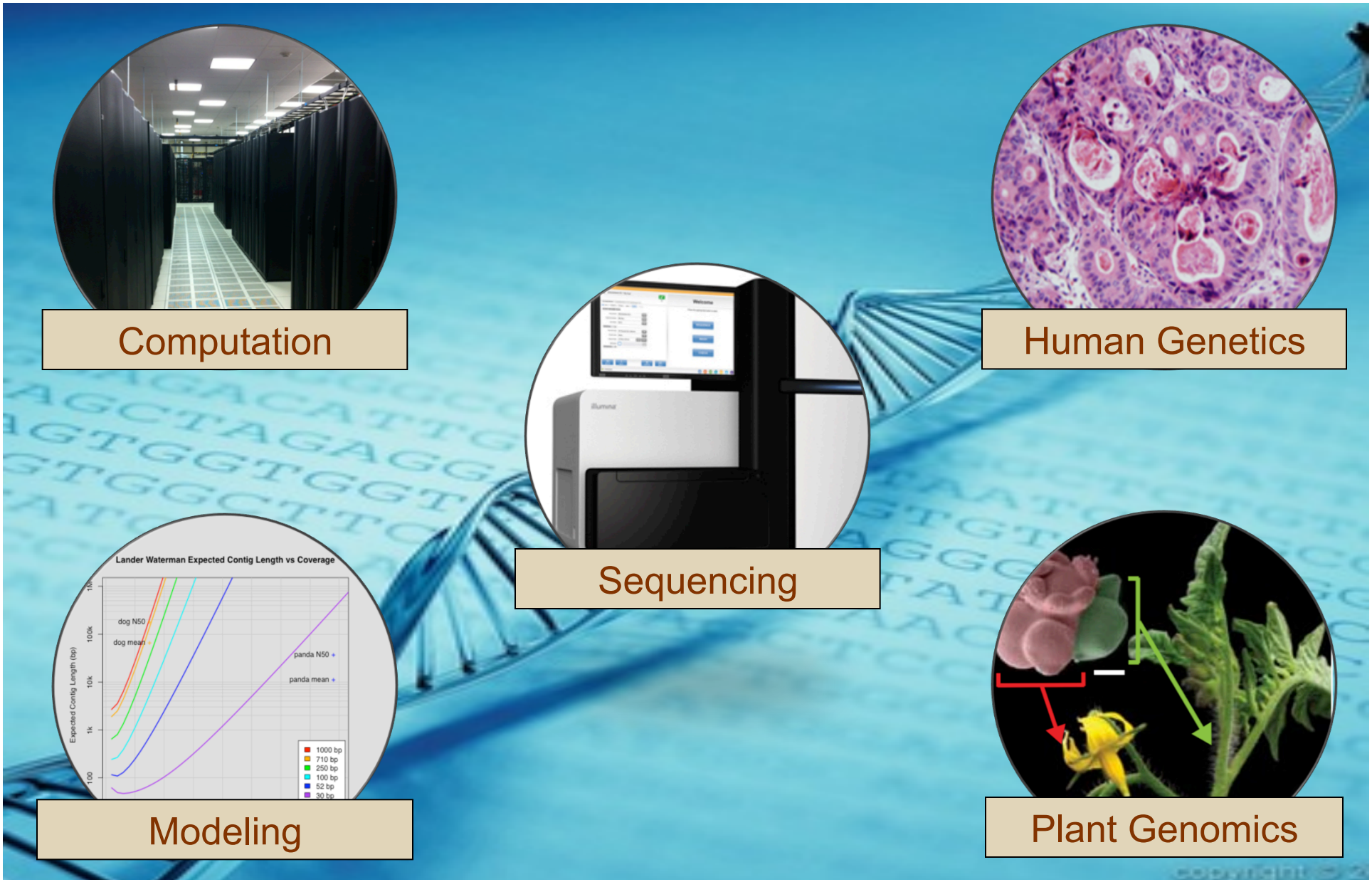
Molecular Biology, Computers & Unix

Michael Schatz

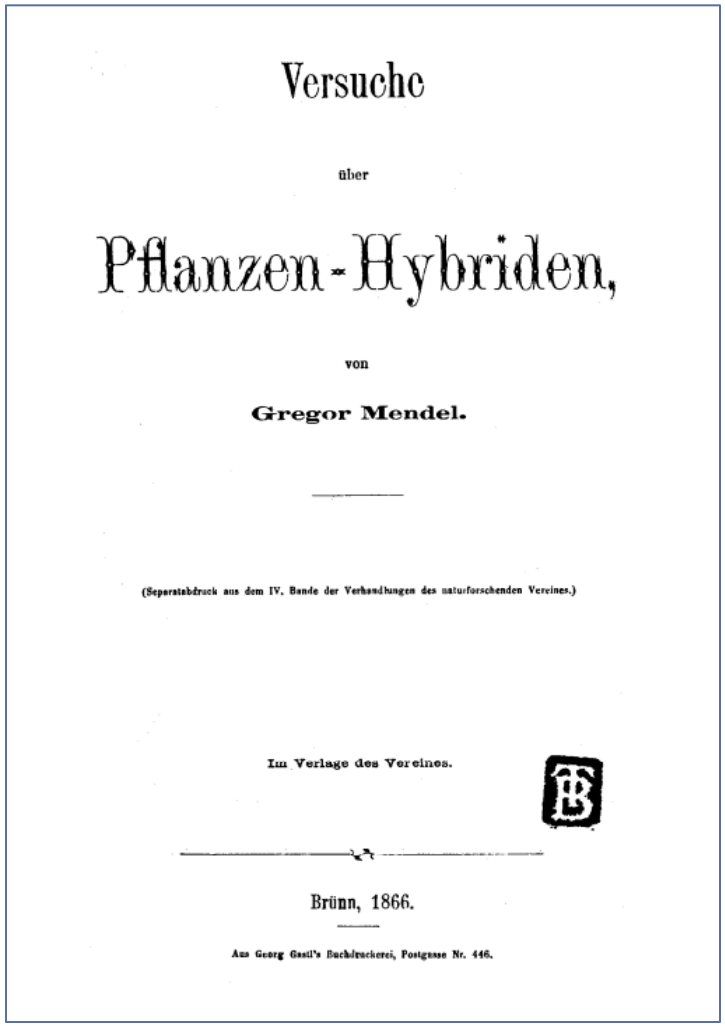
Aug 29, 2012

QB Bootcamp Lecture 1

Schatz Lab Overview



Milestones in Molecular Biology



Versuche

über

Pflanzen-Hybriden,

von

Gregor Mendel.

(Separatabdruck aus dem IV. Bande der Verhandlungen des naturforschenden Vereines.)

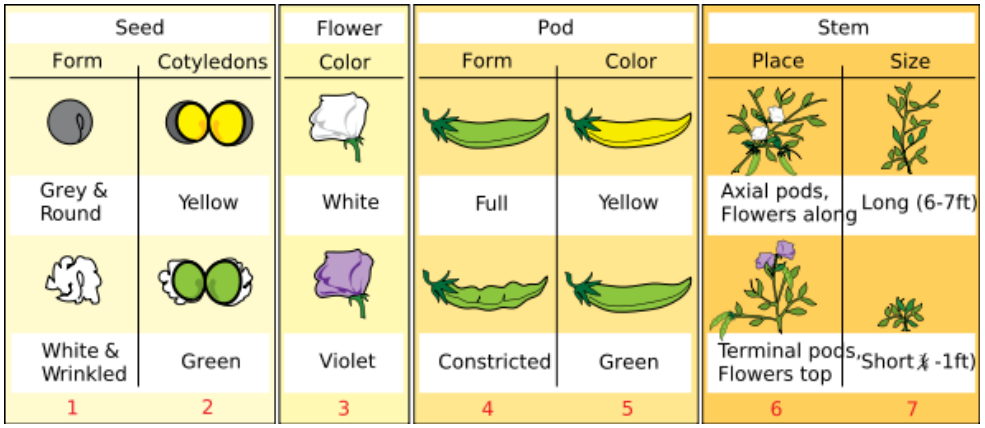
Im Verlage des Vereines.

Brünn, 1866.

Aus Georg Gastl's Buchdruckerei, Postgasse Nr. 446.

Observations of 29,000 pea plants and 7 traits

Generation	*A*	*Aa*	*a*	in Verhältniss gestellt: *A* : *Aa* : *a*
1	1	2	1	1 : 2 : 1
2	6	4	6	3 : 2 : 3
3	28	8	28	7 : 2 : 7
4	120	16	120	15 : 2 : 15
5	496	32	496	31 : 2 : 31
n				2^n-1 : 2 : 2^n-1

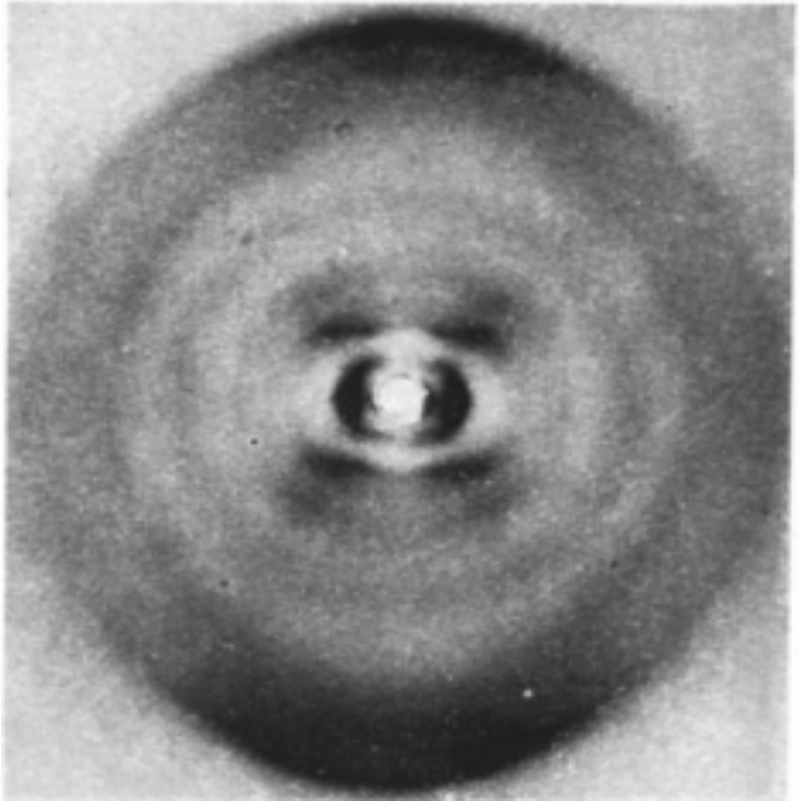


http://en.wikipedia.org/wiki/Experiments_on_Plant_Hybridization

Versuche über Pflanzen-Hybriden. Verh. Naturforsch (Experiments in Plant Hybridization)
Mendel, G. (1866). Ver. Brünn 4: 3–47 (in English in 1901, J. R. Hortic. Soc. 26: 1–32).

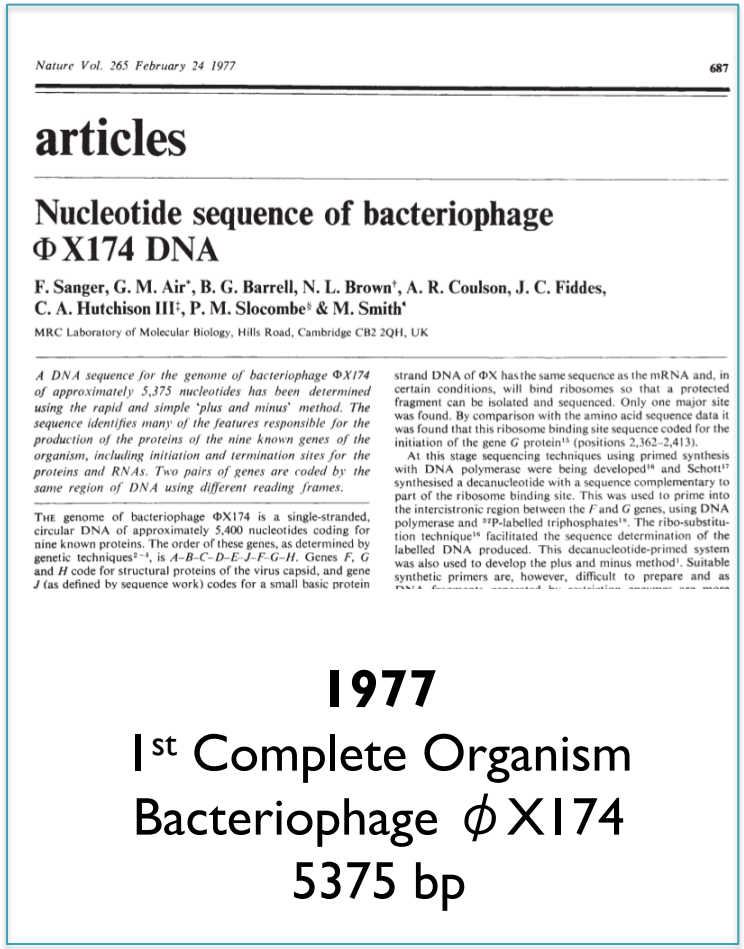
Milestones in Molecular Biology

The origin and behavior of mutable loci in maize
McClintock, B (1950) *Proceedings of the National Academy of Sciences.* 36:344–55.



Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid
Watson JD, Crick FH (1953). *Nature* 171: 737–738.

Milestones in Molecular Biology



Nature Vol. 265 February 24 1977 687

articles

Nucleotide sequence of bacteriophage ΦX174 DNA

F. Sanger, G. M. Air*, B. G. Barrell, N. L. Brown†, A. R. Coulson, J. C. Fiddes, C. A. Hutchison III‡, P. M. Slocombe§ & M. Smith*

MRC Laboratory of Molecular Biology, Hills Road, Cambridge CB2 2QH, UK

A DNA sequence for the genome of bacteriophage ΦX174 of approximately 5,375 nucleotides has been determined using the rapid and simple 'plus and minus' method. The sequence identifies many of the features responsible for the production of the proteins of the nine known genes of the organism, including initiation and termination sites for the proteins and RNAs. Two pairs of genes are coded by the same region of DNA using different reading frames.

THE genome of bacteriophage ΦX174 is a single-stranded, circular DNA of approximately 5,400 nucleotides coding for nine known proteins. The order of these genes, as determined by genetic techniques[1–4], is *A–B–C–D–E–J–F–G–H*. Genes *F*, *G* and *H* code for structural proteins of the virus capsid, and gene *J* (as defined by sequence work) codes for a small basic protein

strand DNA of ΦX has the same sequence as the mRNA and, in certain conditions, will bind ribosomes so that a protected fragment can be isolated and sequenced. Only one major site was found. By comparison with the amino acid sequence data it was found that this ribosome binding site sequence coded for the initiation of the gene *G* protein[15] (positions 2,362–2,413).

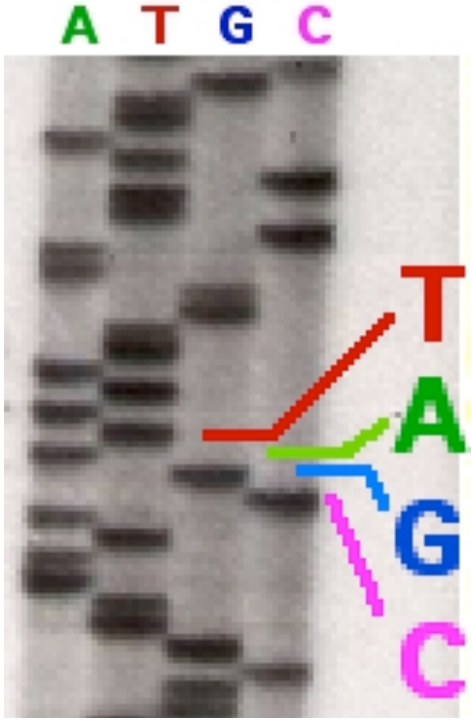
At this stage sequencing techniques using primed synthesis with DNA polymerase were being developed[16] and Schott[17] synthesised a decanucleotide with a sequence complementary to part of the ribosome binding site. This was used to prime into the intercistronic region between the *F* and *G* genes, using DNA polymerase and ^{32}P-labelled triphosphates[18]. The ribo-substitution technique[16] facilitated the sequence determination of the labelled DNA produced. This decanucleotide-primed system was also used to develop the plus and minus method[1]. Suitable synthetic primers are, however, difficult to prepare and as

1977

1st Complete Organism

Bacteriophage ϕX174

5375 bp



Radioactive Chain Termination

5000bp / week / person

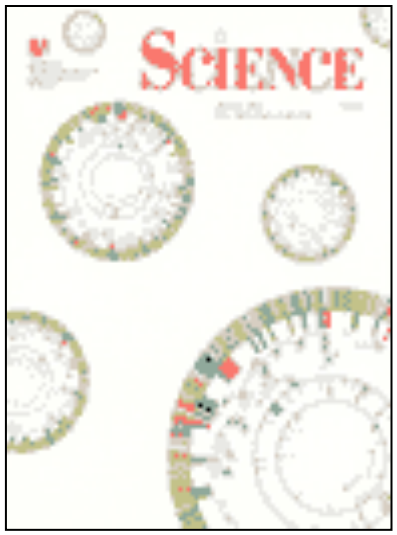
http://en.wikipedia.org/wiki/File:Sequencing.jpg

http://www.answers.com/topic/automated-sequencer

Nucleotide sequence of bacteriophage ϕX174 DNA

Sanger, F. et al. (1977) *Nature. 265: 687 - 695*

Milestones in Molecular Biology



1995
Fleischmann *et al.*
1st Free Living Organism
TIGR Assembler. 1.8Mbp

2000
Myers *et al.*
1st Large WGS Assembly.
Celera Assembler. 116 Mbp

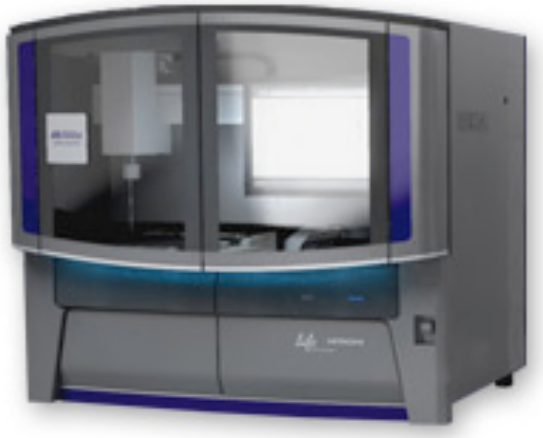
2001
Venter *et al.* / IHGSC
Human Genome
Celera Assembler. 2.9 Gbp

ABI 3700: 500 bp reads x 768 samples / day = 384,000 bp / day.
"The machine was so revolutionary that it could decode in a single day the same amount of genetic material that most DNA labs could produce in a year. " J. Craig Venter

Milestones in Molecular Biology

2004
454/Roche
Pyrosequencing
Current Specs (Titanium):
1M 400bp reads / run =
1Gbp / day

2007
Illumina
Sequencing by Synthesis
Current Specs (HiSeq 2000):
2.5B 100bp reads / run =
60Gbp / day



2008
ABI / Life Technologies
SOLiD Sequencing
Current Specs (5500xl):
5B 75bp reads / run =
30Gbp / day

Milestones in Molecular Biology

There is tremendous interest to sequence:

- What is your genome sequence?
- How does your genome compare to my genome?

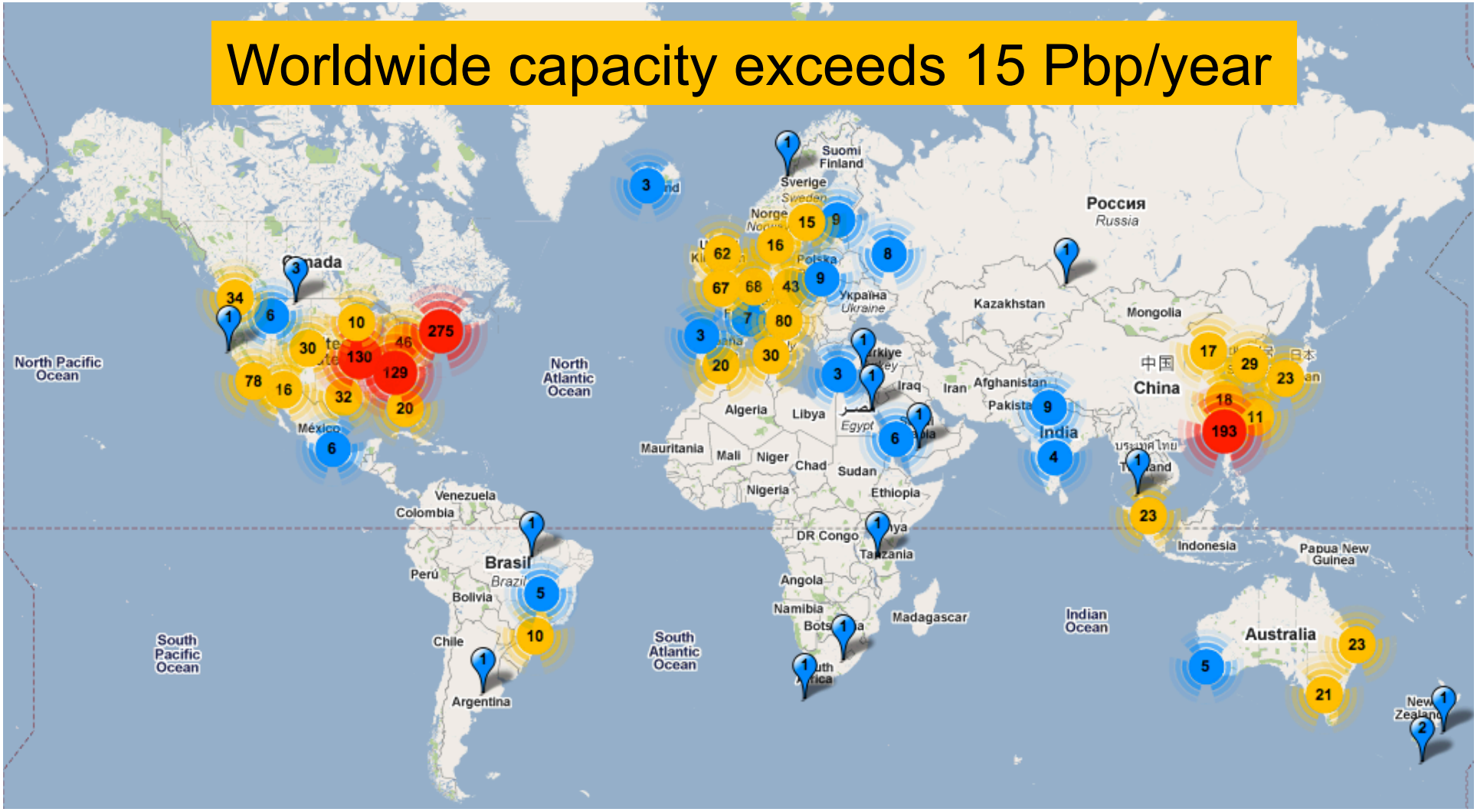
- Where are the genes and how active are they?
- How does gene activity change during development?
- How does splicing change during development?

- How does methylation change during development?
- How does chromatin change during development?
- How does is your genome folded in the cell?
- Where do proteins bind and regulate genes?

- What virus and microbes are living inside you?
- How has the disease mutated your genome?
- What drugs should we give you?

- …

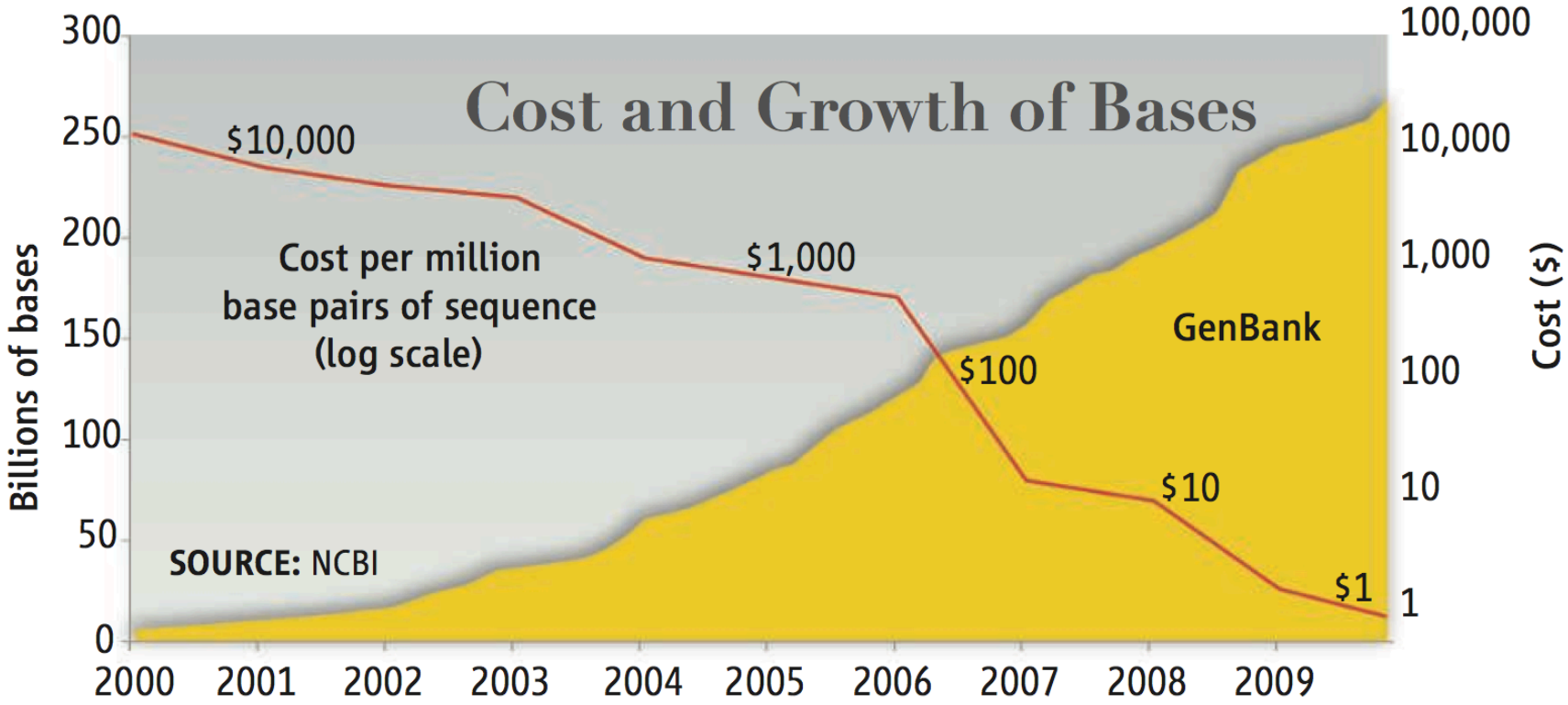
Sequencing Centers



Next Generation Genomics: World Map of High-throughput Sequencers
http://pathogenomics.bham.ac.uk/hts/

DNA Data Tsunami

Sequencing capacity is growing at ~5x per year!
Similar exponential rises across biology: Imaging, mass spec, spike trains, etc…



"Will Computers Crash Genomics?"
Elizabeth Pennisi (2011) *Science*. 331(6018): 666-668.

Modern Biology Challenges

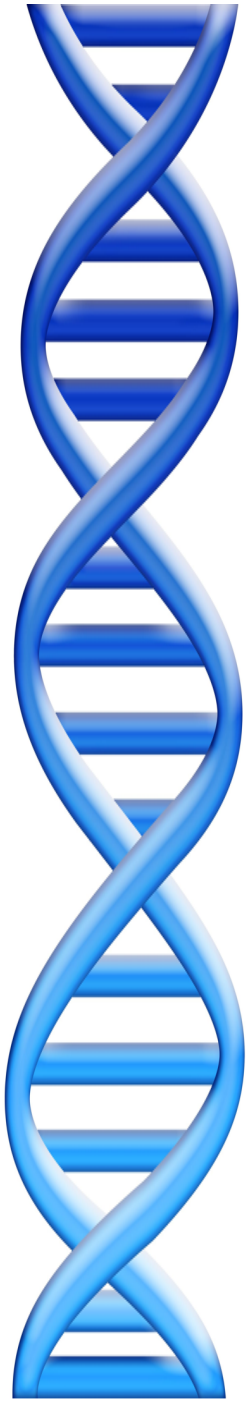
The foundations of biology will continue to be *observation*, *experimentation*, and *interpretation*

- 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



Outline

Part 1: Overview & Fundamentals

- Overview of Computer Systems
- Unix and Scripting Primer

Part 2: Sequence Analysis Theory

Part 3: Genomics Resources

Part 4: Example Analysis

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

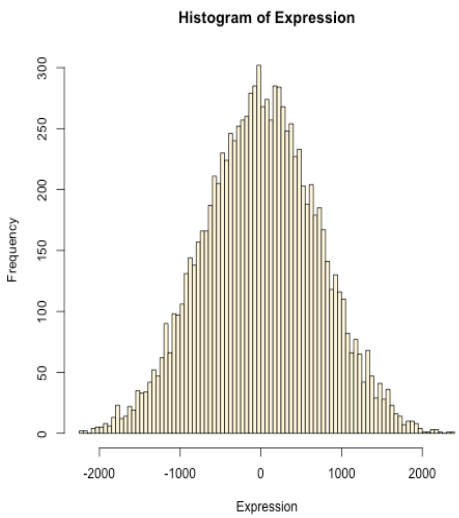
- Modeling & Predictions: How will X respond to Y?

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

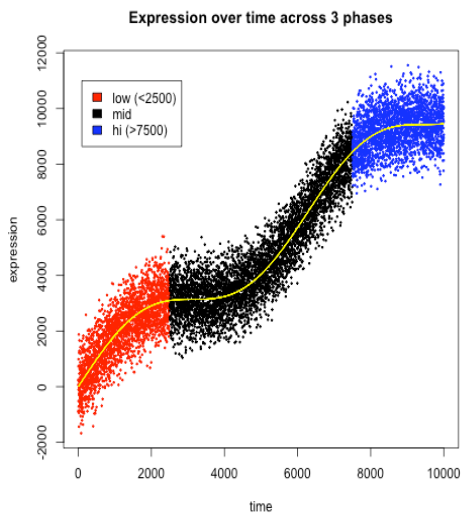
How do we DRAW conclusions?

```
-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
    …
```

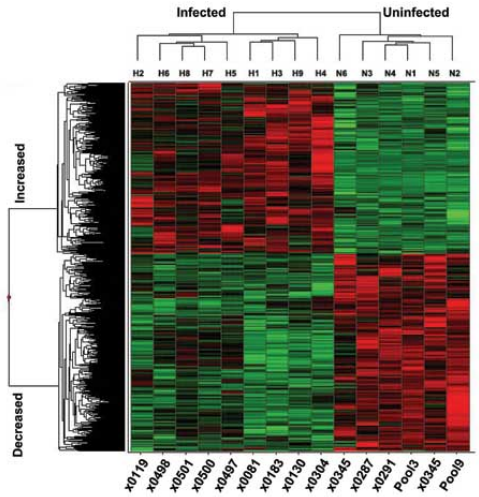
Histogram



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

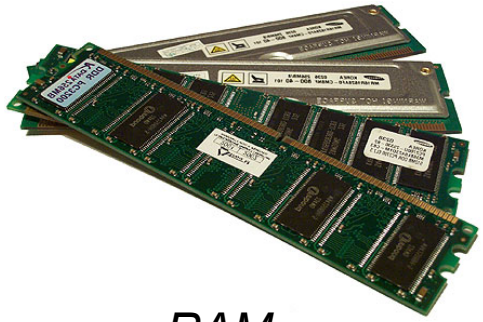
==> HiSeq 2000: 600B observations, 10 days

==> Make friends with your computational tools

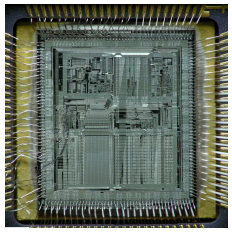
What is a computer?

[hardware]

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



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



Processor
Arithmetic, logic
cores, clock speed

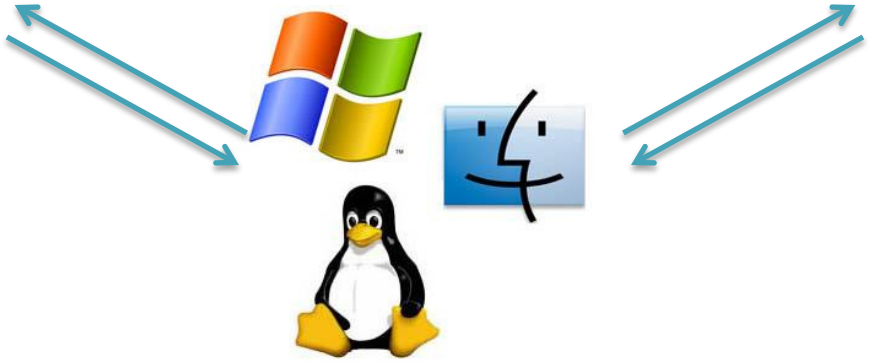
Display
Human Interface

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

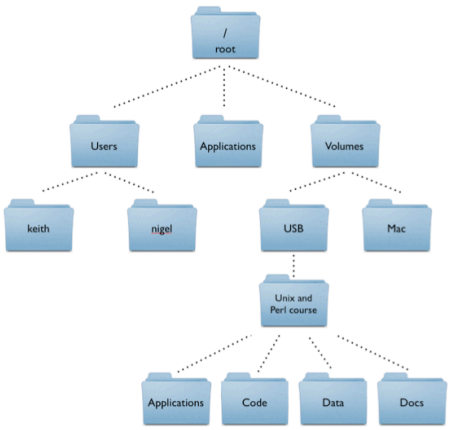
What is a computer?

[software]

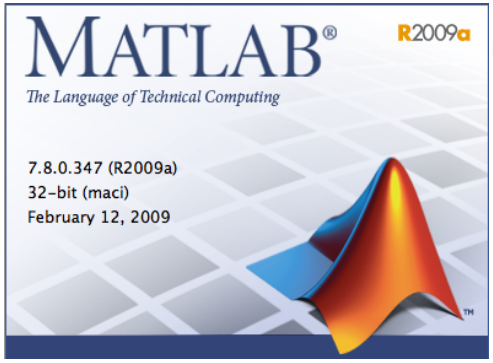
Office Applications
Presentations, Documents
Simple statistics and plots



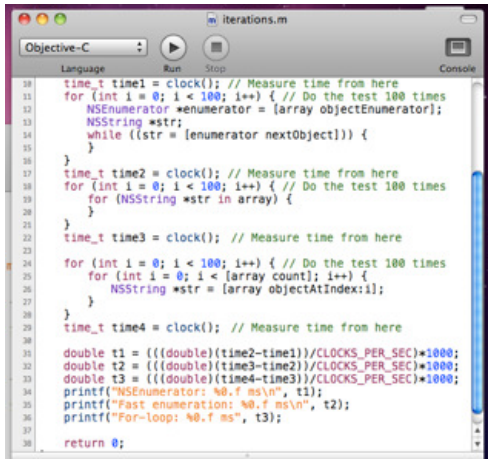
Operating System
Mission Control
Windows, Mac, Unix, iOS



Files / Data
Papers, sequences,
measurements



Scientific Applications
Specialized Analysis
Commercial

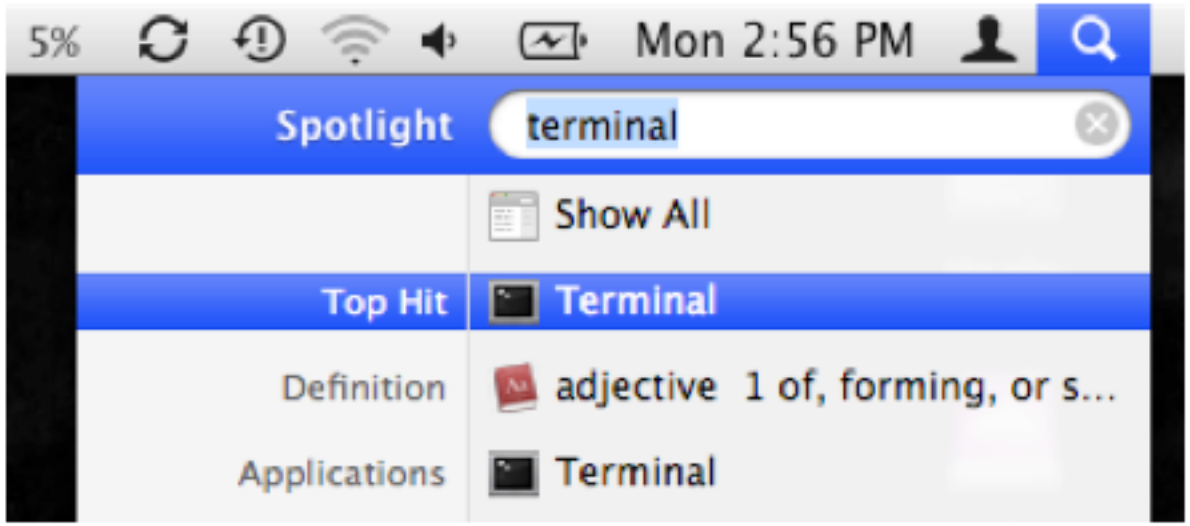


Code / Scripts
Research Applications
Academic

How does scientific software operate?

- The software we need to run is very specialized, there is no 'analyze genome' button in Excel
  - Data files are huge, so probably wouldn't want one anyways
- It takes a lot of work (and time/money) to create a graphical interface to software, so most scientific software uses a 'command line' interface
  - Important to become comfortable using command line tools
- Scientific analyses tend to use workflows consisting of several applications where the output of one phase becomes the input to the next
  - Develop a workflow for dataset X, apply again to dataset Y

Where is the command line?



- Your Mac has a very powerful command line interface hidden just beneath the graphical environment
  - This command line interface is (basically) the same as that used by our scientific cluster BlueHelix
  - Big data files are stored on our central storage system BlueArc

- This environment has a universe of programs you can use to manipulate files and data in novel ways
  - Learning to use this environment is a lot like learning a new language
  - http://korflab.ucdavis.edu/Unix_and_Perl/index.html

File Hierarchy

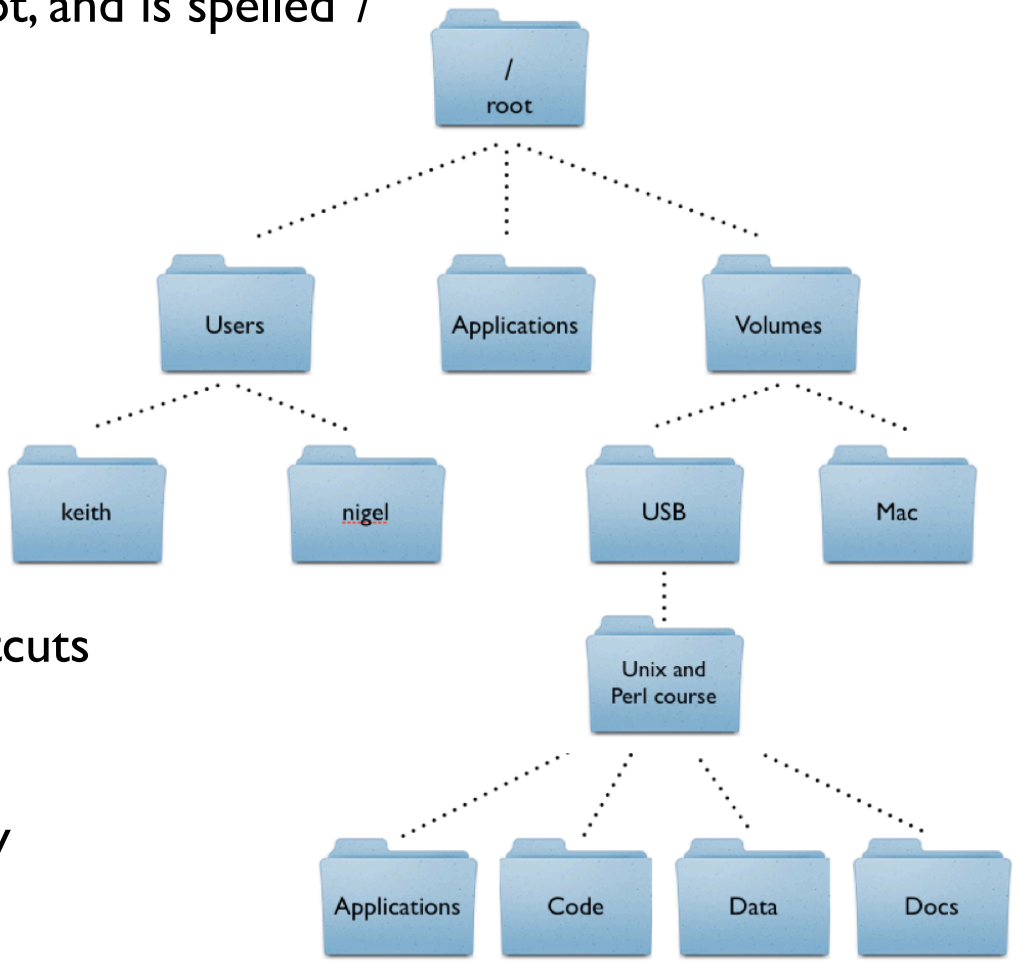
Files are stored in nested directories (folders) that form a tree

- The top of the tree is called the root, and is spelled '/'

- Your home directory (on mac) is at
  /Users/username

- Command line tools are at
  /bin/
  /usr/bin/
  /usr/local/bin/

- A few special directories have shortcuts
  ~ = home directory
  ~bob= bob's home directory
  . = current working directory
  .. = parent directory
  - = last working directory



Working with the shell

- The shell is interactive and will attempt to complete your command as soon as you press enter

```
$ pwd
/Users/mschatz

$ echo “Hello, World”
Hello, World
```

- Here are a few shortcuts that will make your life easier

Command	Effect
Left/Right arrow	Edit your current command
Up/Down arrow	Scroll back and forth through your command history
Control-r	Search backwards through your command history
history	What commands did I just run?
Control-c	Cancel the command
Control-u	Clear the current line
Control-a, Control-e	Jump to the beginning and end of the line

Working with files and directories

- Create directories and copies of the working files

```
$ mkdir myfiles
$ cd myfiles/
$ cp ../At_* .
$ ls -l
total 111648
-rw-r--r--@ 1 mschatz  staff  39322356 Nov  8 01:37 At_genes.gff
-rw-r--r--@ 1 mschatz  staff  17836225 Nov  8 01:37 At_proteins.fasta
```

- Rename files

```
$ mv At_genes.gff Arabidopsis_genes.gff
```

- See how long the files are

```
$ wc -l *
  531497 Arabidopsis_genes.gff
  214021 At_proteins.fasta
  745518 total
```

- Clean up

```
$ cd ..
$ rm -rf myfiles/
```

WARNING!!! Always double check rm

Editing Files

- You can open files from the shell using "regular" applications by their extension

```
$ cp At_genes.gff At_genes.gff.txt
$ open At_genes.gff.txt
$ open .
$ open /Applications/Microsoft\ Office\ 2011/Microsoft\ Word.app/
```

- It is often helpful (or necessary) to edit files within the terminal

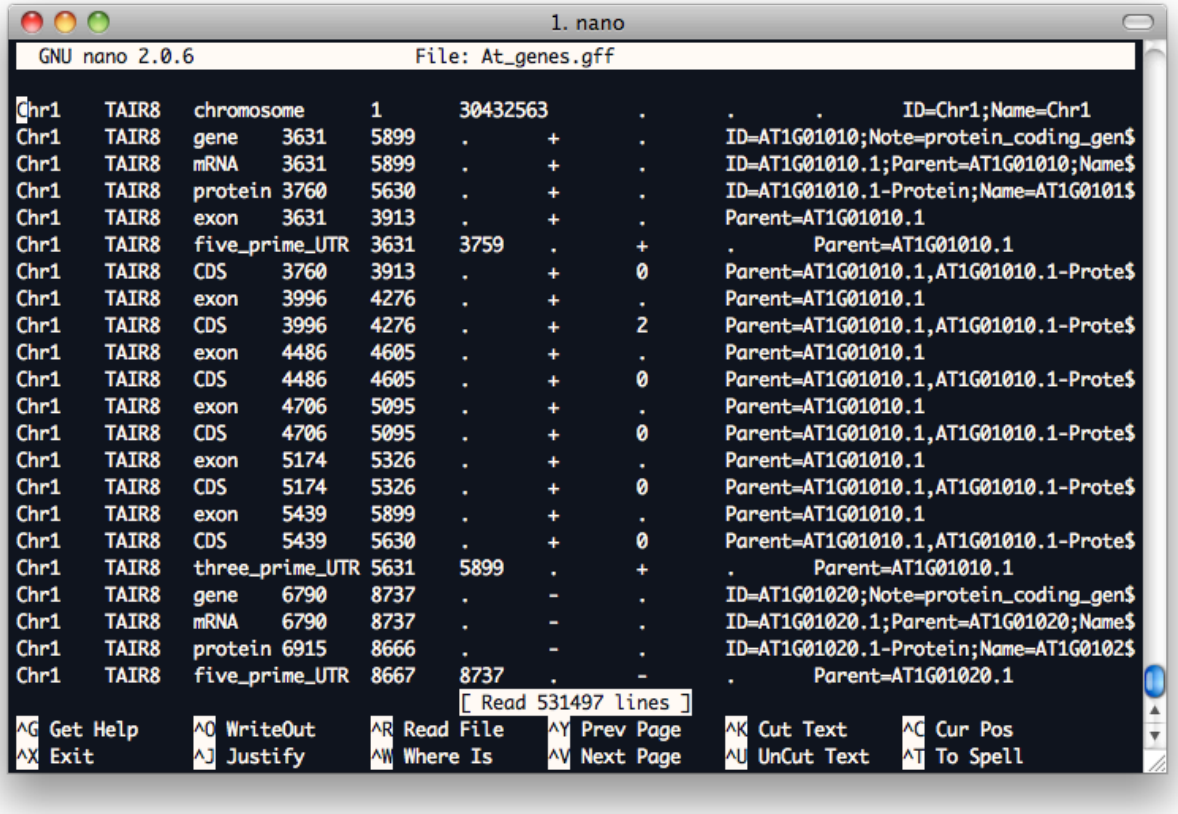
```
$ nano At_genes.gff
```

Basic nano commands

- Type to make edits
- Arrows to move
- Control-O to save
- Control-X to exit
- Control-G for help

Advanced text editors:

- vi
- emacs



Working with text files

- Display the first few lines of a file

```
$ head -5 At_proteins.fasta
>AT1G51370.2 | Symbols:  | F-box family protein | chr1:19049283-19050416 FORWARD
MVGGKKKTKICDKVSHEEDRISQLPEPLISEILFHLSTKDSVRTSALSTKWRYLWQSVPGLDLDPYASSNTNTIVSFVES
FFDSHRDSWIRKLRLDLGYHHDKYDLMSWIDAATTRRIQHLDVHCFHDNKIPLSIYTCTTLVHLRLRWAVLTNPEFVSLP
CLKIMHFENVSYPNETTLQKLISGSPVLEELILFSTMYPKGNVLQLRSDTLKRLDINEFIDVVIYAPLLQCLRAKMYSTK
NFQIISSGFPAKLDIDFVNTGGRYQKKKVIEDILIDISRVRDLVISSNTWKEFFLYSKSRPLLQFRYISHLNARFYISDL
```

- Show the first few proteins names in the file

```
$ grep '>' At_proteins.fasta | head -5
>AT1G51370.2 | Symbols:  | F-box family protein | chr1:19049283-19050416 FORWARD
>AT1G50920.1 | Symbols:  | GTP-binding protein-related | chr1:18874223-18876238 FORW
>AT1G36960.1 | Symbols:  | similar to unknown protein [Arabidopsis thaliana] (TAIR:A
>AT1G44020.1 | Symbols:  | DC1 domain-containing protein | chr1:16719132-16721096 RE
>AT1G15970.1 | Symbols:  | methyladenine glycosylase family protein | chr1:5486538-5
```

- Count how many proteins are present, excluding hypothetical proteins

```
$ grep '>' At_proteins.fasta | wc -l
   32825
$ grep '>' At_proteins.fasta | grep -v 'hypothetical' | wc -l
   31267
```

Working with text files 2

- Create a file of just hypothetical proteins

```
$ grep '>' At_proteins.fasta | grep 'hypothetical' > hypotheticals
$ wc -l hypotheticals
    1558 hypotheticals
```

- Count hypotheticals per chromosome

```
$ cut -f4 -d'|' hypotheticals | head -3
 chr1:11437249-11439801 FORWARD
 chr1:5167349-5168146 REVERSE
 chr1:16717096-16717944 FORWARD

$ cut -f4 -d'|' hypotheticals | cut -f1 -d':' | head -3
 chr1
 chr1
 chr1

$ cut -f4 -d'|' hypotheticals | cut -f1 -d':' | sort | uniq -c
 382  chr1
 234  chr2
 260  chr3
 204  chr4
 384  chr5
   9  chrC
  84  chrM
   1 CAB12631.1 (PTHR11061
```

What happened here?

Working with compressed archives

- Data files are huge! Compress them with gzip to save space

```
$ ls -l At_genes.gff
-rw-r--r--@ 1 mschatz  staff  39322356 Jul  9  2009 At_genes.gff

$ gzip At_genes.gff
$ ls -l At_genes.gff.gz
-rw-r--r--@ 1 mschatz  staff  4601740 Jul  9  2009 At_genes.gff.gz
$ echo "scale=4; 1-4601740/39322356" | bc
.8830
```

Save 88% of the space!

```
$ gzcat At_genes.gff.gz | grep –c mRNA
$ gunzip At_genes.gff.gz
```

Stream through the compressed file
Or unzip it

- Use tar to compress and bundle a set of files

```
$ du -h Arabidopsis/
 95M Arabidopsis/
$ tar czvf Arabidopsis.tar.gz Arabidopsis/
$ ls -lh Arabidopsis.tar.gz
-rw-r--r--  1 mschatz  staff    25M Aug 27 14:27 Arabidopsis.tar.gz
$ tar xzvf Arabidopsis.tar.gz
```

du recursively prints the sizes of directories

Save 73% of the space!

Monitoring Processes

- Unix systems can run many commands and by many users at once
  - Especially useful for commands that run for a long time
  - Especially useful for servers that have special resources

```
$ ps
  PID TTY           TIME CMD
60820 ttys000    0:00.30 /bin/bash

$ ps aux | head -3
USER       PID  %CPU %MEM      VSZ    RSS   TT  STAT STARTED      TIME COMMAND
root     21527   1.7  0.1  3129268   5692   ??  Ss   11Jul12 679:00.75 /
Library/Application Support/iStat local/iStatLocalDaemon
mschatz  62928   1.6  1.4  2986576 119648   ??  S    31Jul12 895:05.37 /
System/Library/CoreServices/SystemUIServer.app/Contents/MacOS/SystemUIServer
```

- Monitor use of the system

```
$ top
(press q to quit)
```

Background Processes

- Any number of processes can run in the background
  - Use the ampersand (&) to launch a process into the background
  - Alternatively use control-z to pause a process, then use 'bg'

```
$ du -a /
(control-c to cancel)

$ du -a / | sort -nrk1 > ~/filesizes.txt
(control-z to stop)
$ bg
$ du –a / | sort –nrk1 > ~/filesizes.txt.2 &
```

- List running jobs associated with this shell

```
$ jobs
$ fg %1
(control-z to stop)
$ bg
```

- Kill off run-away commands

```
$ ps
$ kill 61110
$ kill -9 61110
```

61110 is the process id I want to kill
kill -9 for really stubborn processes

Working with remote servers

- Use SSH to connect to a remote server

```
$ ssh mschatz@bhdev1.cshl.edu
```

- The server runs UNIX, and the standard commands are available

```
$ ls –l | sort –nrk5 | head -3
$ who
```

- There are special lab directories for CSHL users (> 1PB of storage total)

```
$ df –h /data/schatz* /data/wig*
```

- Your lab may have special commands available

```
$ ls /data/schatz/software/bin/
$ /data/schatz/software/bin/samtools
```

- Typing out the full path for every command is a pain, edit your bashrc

```
$ nano ~/.bashrc
```

(at the bottom add: export PATH=~/bin:/data/schatz/software/bin/:$PATH)
Control-o to save

See: http://intranet.cshl.edu/it/bluehelix/ for details on the shared cluster

Files and permissions

- Every file has an owner and a group, you can only read/write to a file if you have permission to do so

```
$ pwd
/Users/mschatz/Desktop/Unix_and_Perl_course/Data/Arabidopsis

$ ls -l
total 193976
-rw-r--r--@ 1 mschatz  staff  39322356 Jul  9  2009 At_genes.gff
-rw-r--r--@ 1 mschatz  staff  17836225 Oct  9  2008 At_proteins.fasta
-rw-r--r--@ 1 mschatz  staff  30817851 May  7  2008 chr1.fasta
-rw-r--r--@ 1 mschatz  staff  11330285 Jul 10  2009 intron_IME_data.fasta
```

- These files can be read by anyone, but only written by me
  - Change permissions with 'chmod'

```
$ chmod g+w At_*
$ man chmod
```

- Programs and scripts have the execute bit set

```
$ ls -l /bin/ls
-r-xr-xr-x  1 root  wheel  80688 Feb 11  2010 /bin/ls*
```

Programming Basics: Loops

- A bash script is just a list of commands

```
$ cat simple_script.sh
#!/bin/sh

echo "Hello, World”
echo "Shall we play a game?”

$ chmod +x simple_script.sh
$ ./simple_script.sh
```

[What does this do?]

Programming Basics: Loops

- A bash script is just a list of commands

```
$ cat simple_script.sh
#!/bin/sh

echo "Hello, World”
echo "Shall we play a game?”

$ chmod +x simple_script.sh
$ ./simple_script.sh
```

[What does this do?]

- Things get interesting when we add variables and loops

```
$ cat loop_script.sh
#!/bin/sh

for name in "Mike" "Justin" "Mickey"
do
  echo "Hello, $name" >> authors.txt
  everyone="$name $everyone"
done
echo "Hello: $everyone" >> authors.txt

$ chmod +x loop_script.sh
$ ./loop_script.sh
$ ./loop_script.sh
$ ./loop_script.sh
```

Use >> to append

[What does this do?]

Programming Basics: Conditionals

- Conditionals and loops let us work over any number and type of file

```
$ cat conditional_script.sh
#!/bin/sh

for filename in `/bin/ls * | grep –v ".sh"`
do
  type=`echo $filename | cut -f2 -d'.'`
  echo "Processing $filename, type is $type"
  echo "=================="

  if [[ $type == "fasta" ]]
  then
    protein_count=`grep -c '>' $filename`
    hypo_count=`grep -c hypothetical $filename`
    echo "$filename has $protein_count proteins, $hypo_count are hypothetical"
  elif [[ $type == "gff" ]]
  then
    echo "$filename stats"
    cut -f3 $filename | sort | uniq -c
  else
    echo "Unknown file type"
  fi

  echo "=================="
  echo
done
```

The backtics `<cmd>`
Let us run commands
inside of other commands

[What does this do?]

Programming Basics: Arguments

- The shell defines a few special variables to specify input

```
$ cat argument_script.sh
#!/bin/sh

if [[ $# -lt 2 ]]
then
  echo "USAGE: argument_script.sh proteinsfile type_1 .. type_n"
  exit
fi

echo "Script was run as: $0"
echo "First argument is: $1"
echo "Second argument is: $2"

proteinsfile=$1
shift

while [ $# -gt 0 ]
do
  type=$1
  shift
  count=`grep '>' $proteinsfile | grep -c $type`
  echo "There are $count $type proteins in $proteinsfile"
done

$ ./argument_script.sh At_proteins.fasta F-box GTP-binding hypothetical
```

$# stores number of arguments

$0 has script name
$1-$9 have first 9 arguments

Use shift to access arguments

Loop until there are no more types to consider

Programming Basics: Functions

- A function is a reusable block of code

```
$ cat function_script.sh
#!/bin/sh

function log()
{
  date=`date`
  echo "$date :: $*"
}

function processFasta()
{
  file=$1
  log "Processing fasta: $file"
  num=`grep -c '>' $file`
  log "There are $num sequences"
}

function processGFF()
{
  file=$1
  log "Processing gff: $file"
  num=`wc -l $file`
  log "There are $num records"
}

for file in `/bin/ls *`
do
  log "Processing $file"

  type=`basename $file | cut -f 2 -d'.'`

  if [[ $type == "fasta" ]]
  then
    processFasta $file
  elif [[ $type == "gff" ]]
  then
    processGFF $file
  else
    log "Unknown filetype $type"
  fi
done
```

Programming Resources

- Much like learning a new spoken language, computer languages have their own syntax and grammar that will be unfamiliar at first, but get easier and easier over time
  - There are many ways to accomplish the same task
  - You can quickly become a data magician

- The way to learn a new computer language is to practice speaking it
  - The ~30 commands you have seen today can be combined together into an infinite number of combinations
  - Lots of good resources available online:
    - http://www.molvis.indiana.edu/app_guide/unix_commands.html
    - http://tldp.org/LDP/abs/html/index.html
    - http://stackoverflow.com/
    - http://google.com

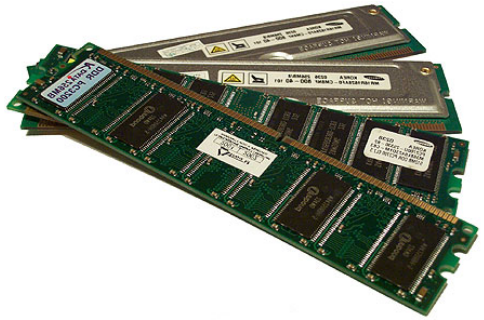
WARNING: Computers are very unforgiving

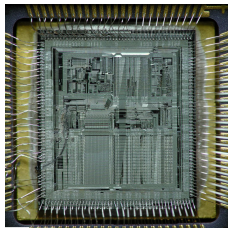
- 'rm –rf /' <= delete every file on your computer
- 'cp junk.doc thesis.doc' <= overwrite your thesis with junk.doc
- 'cat results.partial > results.all' <= oops, should have appended with >>


READING THE ENTIRE GENOME CODE...
Listen to this part: 'AGGCTAATC CGCATAACTG'
Wow!

Break

Hardware review





Unix Review

Command	Output
man	Look up something in the manual (also try Google)
ls	List the files in the current directory
cd	Change to a different directory
pwd	Print the working directory
mv, cp, rm	Move, copy, remove files
mkdir, rmdir	Make or remove directories
cat, less, head, tail, cat	Display (parts) of a text file
echo	Print a string
sort, uniq	Sort a file, get the unique lines
find, grep	Find files named X, or containing X
chmod	Change permissions on a file
wc	Count lines in a file
jot / seq	Output numbers from 1 to X (on Linux use seq)
| (pipe), > (redirect)	Send output to a different program, different file

Programming Review

Variables & Arguments

```
names=Mike
names="$names Justin"
names="$names Mickey"
echo $names

echo "There are $# arguments: $*"
shift
echo "The second argument is $1"
```

Conditionals

```
if [[ $type == "fasta" ]]
then
    num=`grep –c '>' $file`
    echo "There are $num seqs"
elif [[ $type == "gff" ]]
then
    num=`wc -l $file`
    echo "There are $num records"
else
    echo "Unknown file type"
fi
```

Loops

```
rm authors.txt
for name in Mike Justin Mickey
do
    echo $name >> authors.txt
    c=`cat authors.txt | wc -l`
    while [ $c -gt 0 ]
    do
        echo $name $c
        c=`echo $c-1 | bc`
    done
done
```

Functions

```
function log()
{
  date=`date`
  echo "$date :: $*"
}
for name in Mike Justin James
do
  log "Processing $name"
  echo $name >> authors.txt
  log "Done with $name"
done
```

Scripting Challenges

1. Create 1000 files named mutantA.X.txt with X in [1,1000] that contain the numbers 1 to X

   mutantA.1.txt: 1
   mutantA.2.txt: 1 2
   mutantA.3.txt: 1 2 3
   …

2. Rename 1000 files named mutantA.X.txt to mutantB.X.txt?

   mutantA.1.txt => mutantB.1.txt
   mutantA.2.txt => mutantB.2.txt
   mutantA.3.txt => mutantB.3.txt
   …

3. Identify the files in the given directory that contain a specified keyword and copy them to a specified directory

   ./find_special.sh search_directory 976 destination_directory
   => cp search_directory/mutantB.976.txt destination_directory
   => cp search_directory/mutantB.977.txt destination_directory
   => cp search_directory/mutantB.978.txt destination_directory
   ...

Questions?

http://schatzlab.cshl.edu