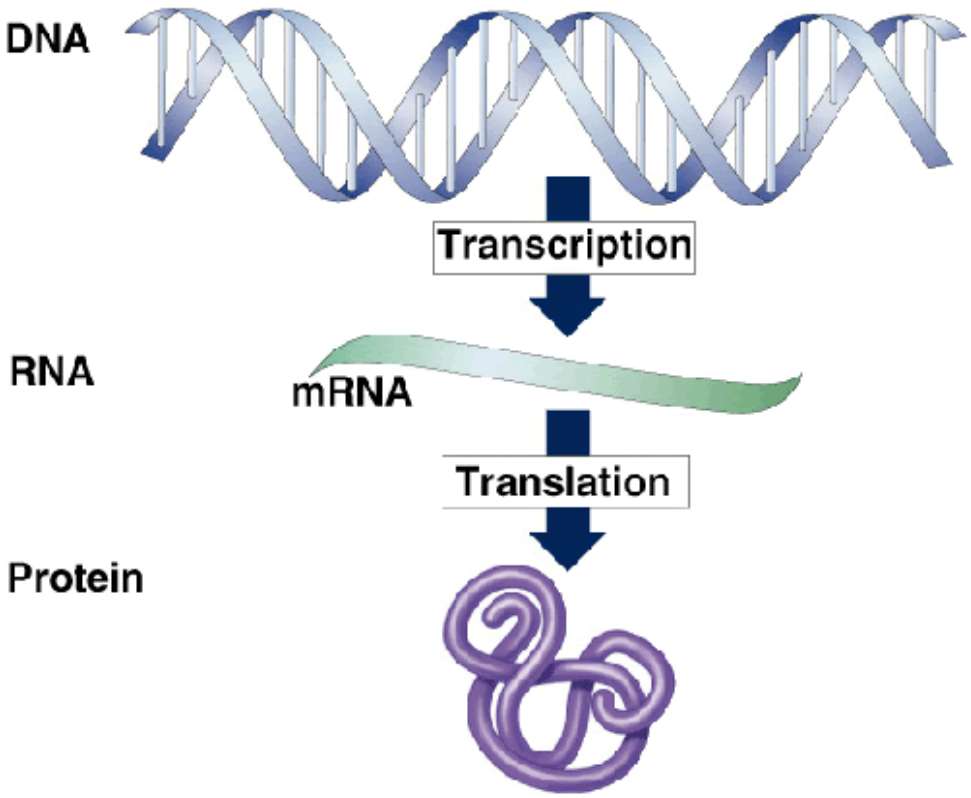
This course should have been named “Computational Genomics” and other stuff for the first day.

Goals

- Introduce bioinformatics as an area of knowledge
- Describe course approach
- Understand why bioinformaticists use Linux and a command line interface

Assumptions of prior knowledge

- Genome
- DNA/RNA
- Nucleotides
- Amino acids
- Transcription
- Translation
- Genotype
- Sequence Reads
- SNPs



Student backgrounds

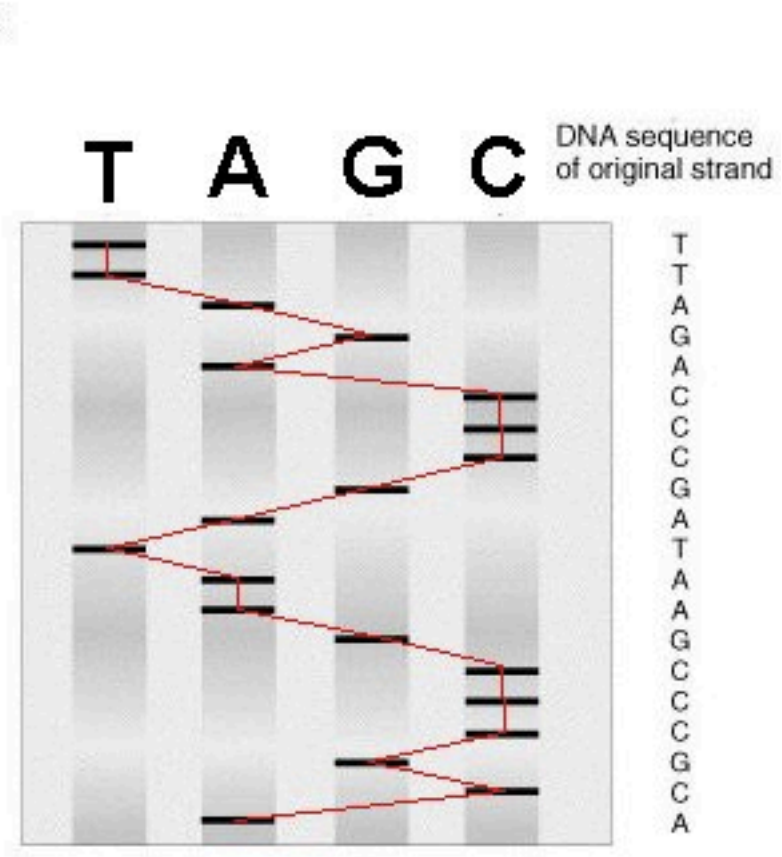
- Some people are more advanced than others
- Speak up when something doesn't make sense
- Be kind to others and help out! Science is all about collaboration.

Bioinformatics

- Deriving biological understanding from large amounts of data with specialized computational skills and tools
- Originally used to refer only to computational *molecular* biology. Now used in a broader sense of any type of biology and computers. (IE ecological data, image data, etc)
- Overlap of many disciplines
  - Computer Science
  - Biology
  - Statistics
  - Applied math
  - Molecular Biology
  - Biochemistry
  - Data Science

Some History

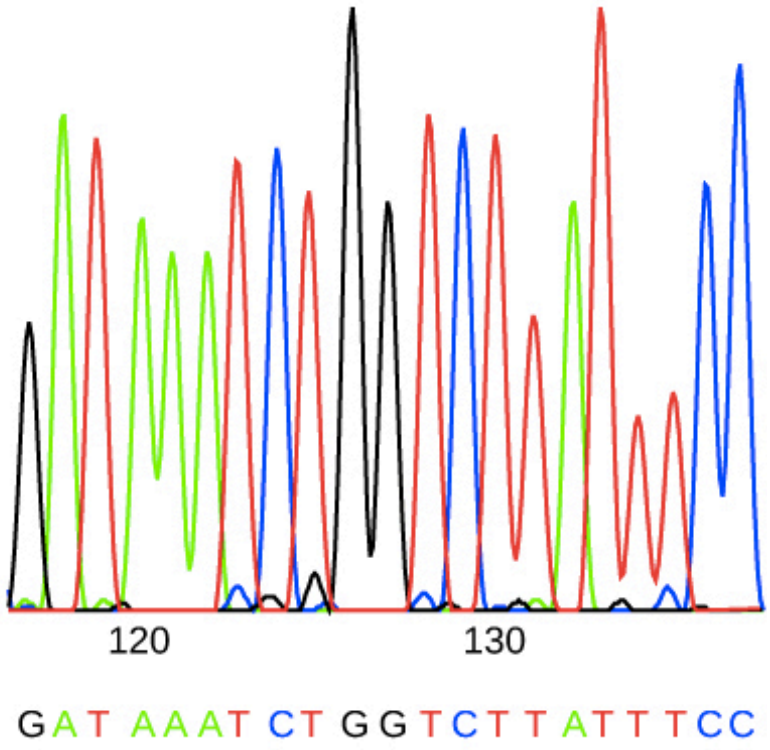
- 1977 – Sanger sequencing
- 30kb of sequence data per day by the 1980s
  - Already too much for a person to analyze!
- Needs
  - Merge small sequence pieces into longer sequences (assembly)
  - Editing sequences
  - Searching (for example, for restriction sites)
  - Translate sequences into amino acids
- First bioinformatics software packages emerged in the late 1970s as well

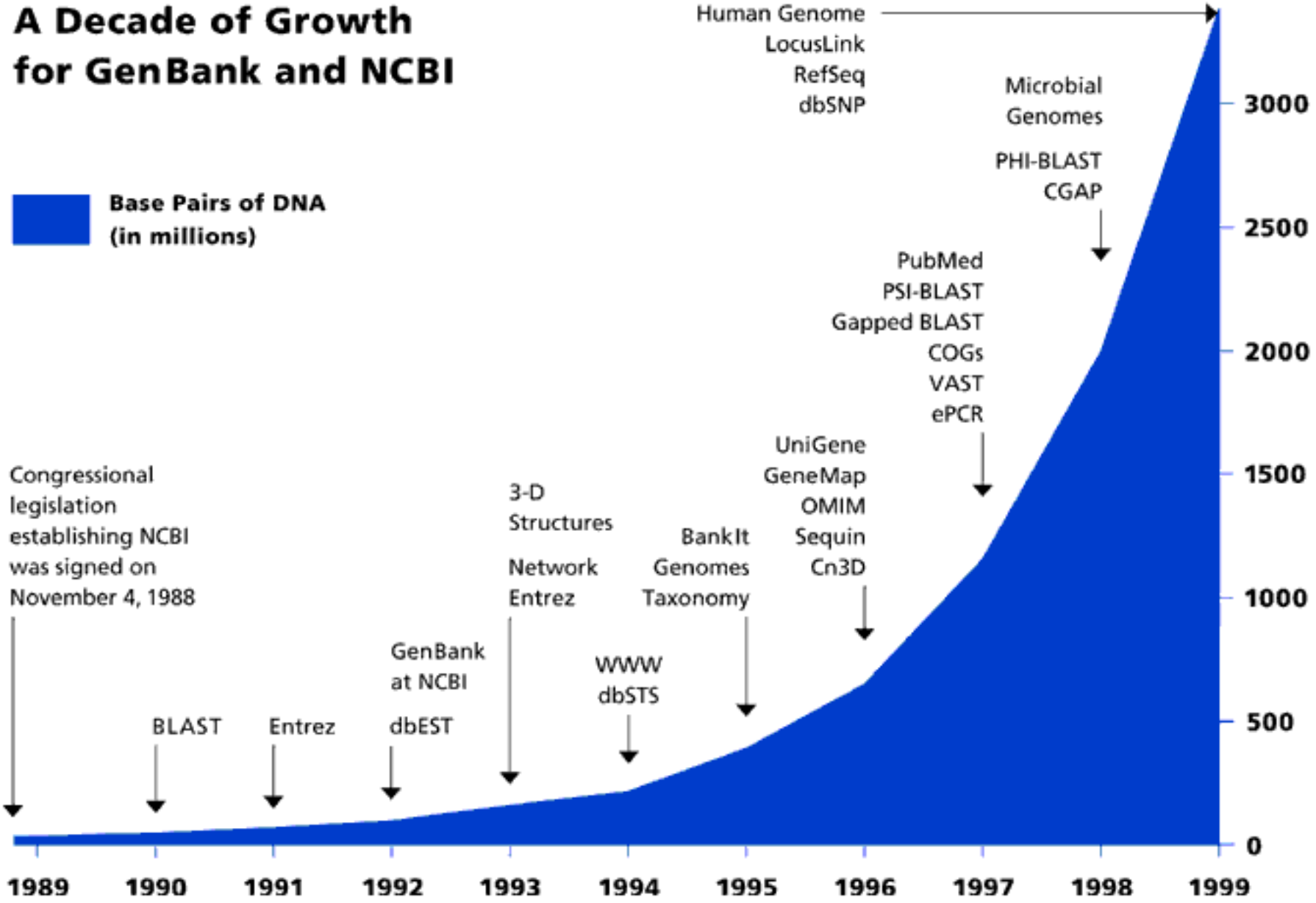


The early days of bioinformatics publishing, Roberts, 2000, Bioinformatics

More history

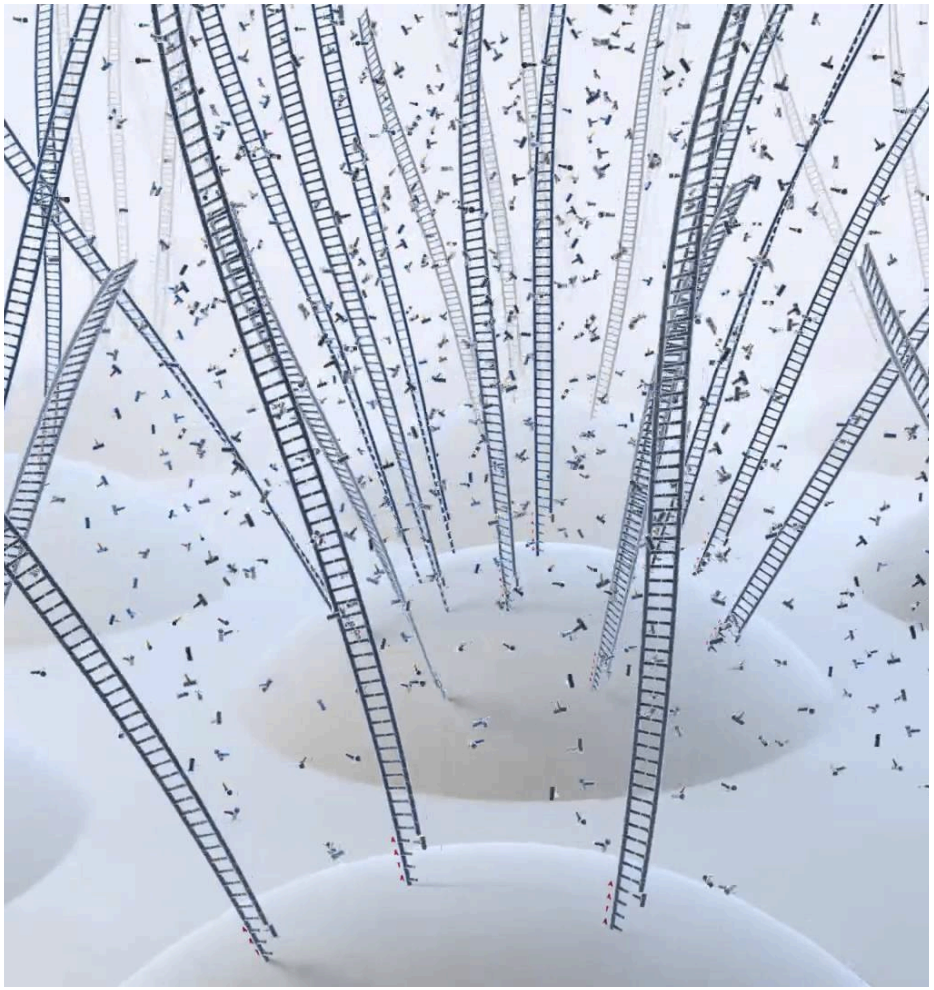
- Sequencing became truly automated in 1996 with ABI machines
- This led to shotgun sequencing
- Assembly of sequenced fragments became a critical concern
- Not a trivial problem due to repeats and sequencing error
- Investment in human genome (1990-2001) was partially an investment in bioinformatics software for assembly, visualization and annotation



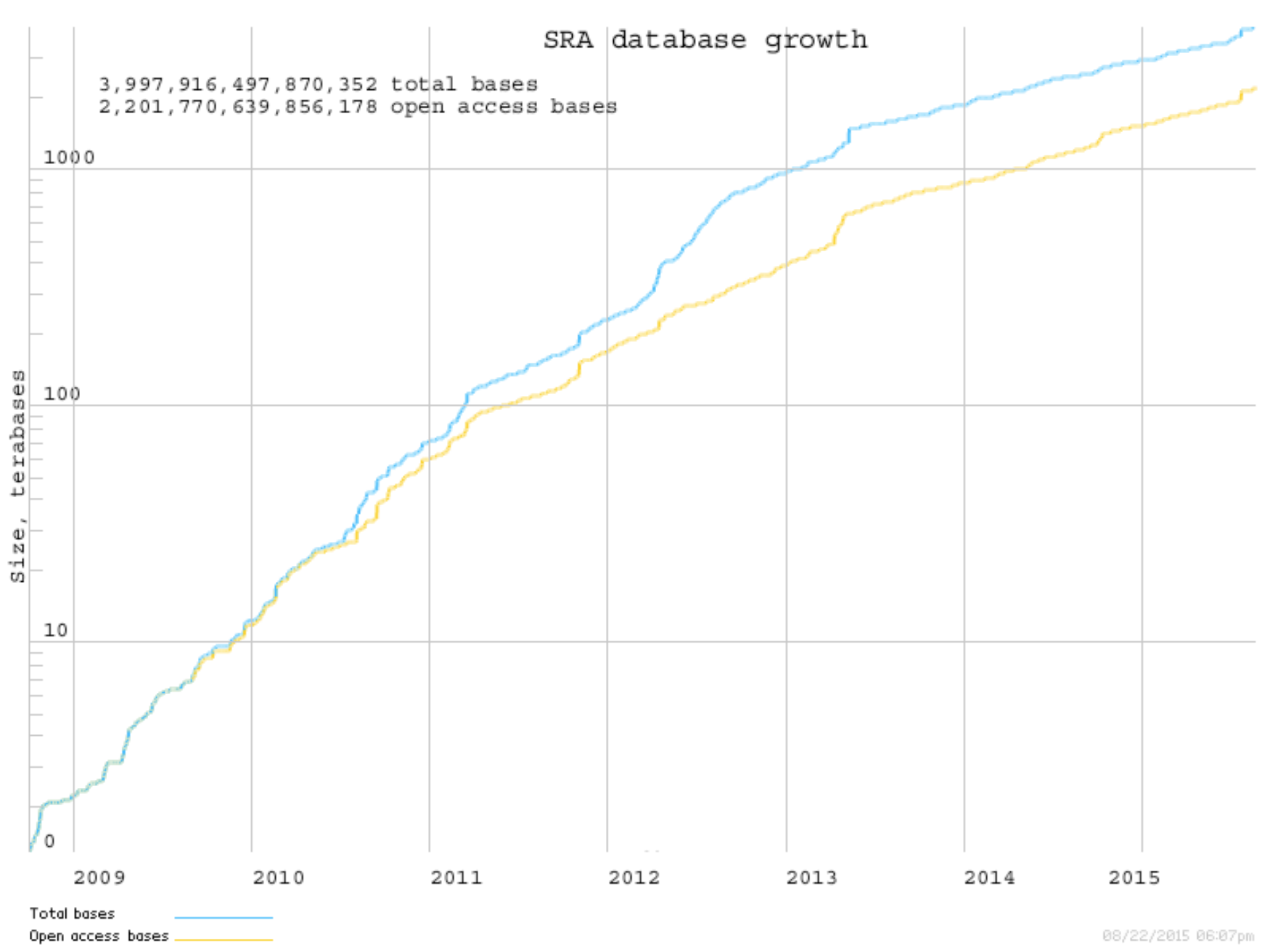

A Decade of Growth for GenBank and NCBI
Base Pairs of DNA (in millions)
Congressional legislation establishing NCBI was signed on November 4, 1988
BLAST
Entrez
GenBank at NCBI
dbEST
3-D Structures
Network Entrez
WWW
dbSTS
BankIt
Genomes
Taxonomy
UniGene
GeneMap
OMIM
Sequin
Cn3D
PubMed
PSI-BLAST
Gapped BLAST
COGs
VAST
ePCR
Microbial Genomes
PHI-BLAST
CGAP
Human Genome
LocusLink
RefSeq
dbSNP
0
500
1000
1500
2000
2500
3000
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999

More history

- Next generation sequencing has changed the field
- Bigger problems – scale up EVERYTHING!
- More applications



From Qiagen video


SRA database growth
3,997,916,497,870,352 total bases
2,201,770,639,856,178 open access bases
1000
100
10
0
Size, terabases
2009
2010
2011
2012
2013
2014
2015
Total bases
Open access bases
08/22/2015 06:07pm

Aims of Bioinformatics

- Organize data to allow access and integration of new data
- Develop tools and resources to analyze data
- Analyze the data and interpret the results in a biologically meaningful manner

From human health to climate change to agriculture, bioinformatics is a critical part of finding solutions.

Luscombe et al., 2001

Bioinformatics vs Computational Biology

- Slightly different, but overlapping
- There are spectrums (as in any field) of applied vs theoretical
  - Studying biology directly (using tools) vs developing tools
  - Computational pipelines and algorithms vs pure mathematics and algorithms
- New terms: quantitative biology, systems biology
- Some people are arguing for “modern biology” or “contemporary biology” to subsume all of these
  - biology is necessarily tied to math and computers
- Continued evolution of discipline is not surprising, it is quite young (with disputed origin of term “bioinformatics” as late as 1991)

http://www.bioinformatics.org/wiki/Origins_of_bioinformatics

Topics

- DNA Sequence
  - Genomics -> Metagenomics
  - Epigenomics
- RNA Sequence
  - Transcriptomics -> Metatranscriptomics
  - Non-coding RNAs
- Microarrays
- Proteomics
- Metabolomics
- Structural bioinformatics
- Comparative genomics
- Machine learning
- Image processing
- Data visualization
- Ontologies
- Databases and knowledge management
- Medical informatics
- Biostatistics
- Phylogenetics
- Data simulation
- Natural language text mining

We will focus on sequence data.

- Why?
- Abundant
- Broad applications across biology
  - variant detection and genotyping
  - transcriptome sequencing for gene expression studies
  - protein-DNA interaction assays like ChIP-seq
  - bisulfite sequencing for methylation studies
- Bioinformatics (especially getting started) is often mostly about data processing skills. Sequencing data is a perfectly good learning substrate for data processing.

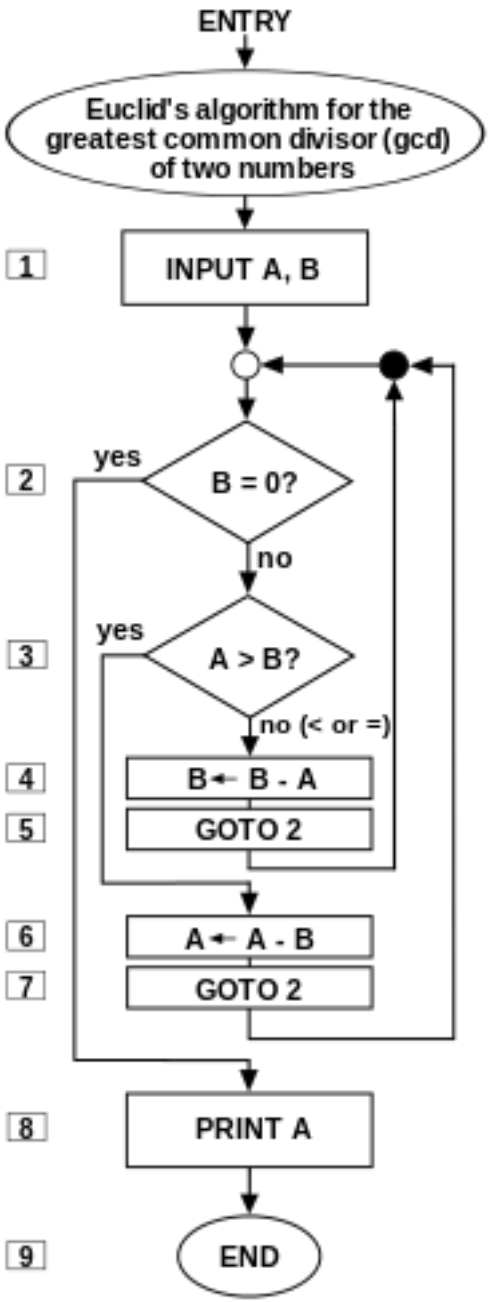
Common problems for high throughput sequencing

- Data quality analysis and trimming
- DNA read alignment
- RNA read alignment
- Statistics of differential gene expression
- *De novo* read assembly
- Variant detection
- Visualization
- File formats and storage

Algorithms

a step by step set of operations to be performed to solve a defined problem

- Steps are built in a formal language
- Development of algorithms is often concerned with efficiency (ie with memory and CPU limitations)
- Can the problem be approximated to be more efficient?



Some Philosophy

- Bioinformatics changes tremendously quickly
- Bioinformatics skills – large scale:
  - Be able to experiment with data on a computer
  - Be able to understand/interpret the results
- Skills on a finer scale
  - Deal with complex, often not well defined, file formats, and how to reformat data
  - Utilize numerous pieces of software (pipelining)
  - Access the right pieces of software and data
  - Submit your data and code to appropriate repositories

Reproducibility

- Reproducibility is the ability to duplicate an entire experiment or study, either by the same researcher or by someone else working independently.
- Also called replicating
- Reproducibility is one of the main principles of the scientific method.
- Scientific validity = Independent replication of experimental results
- In bioinformatics, this requires sharing code, analysis details and raw data
  - Data sharing is usually required by peer-reviewed journals
  - Code/analysis sharing is less standardized, but coming soon
  - See readings

Robustness

- In wet lab research, it is often obvious when an experiment fails.
- This is not always true with computational analysis. Software may not print an error, even when things have gone wrong.
- Fewer users = fewer bug reports
- High dimensional data is complex and difficult to have a priori expectations

Robustness

- Never trust your data or tools – always verify in whatever way possible
- Look at results at each intermediate step
- Visualize output in the most meaningful way possible
- Use good controls and examine them in comparison to treatments often

How do I do reproducible and robust research?

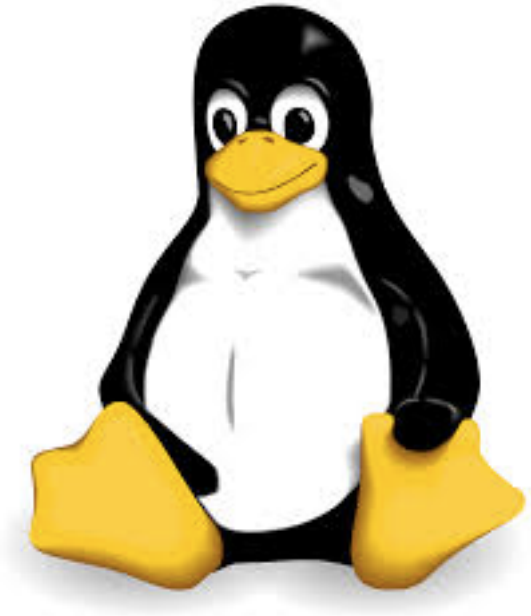
- Record keeping – keep a (preferably online) lab notebook for the dry lab
- Scripting to automate tasks
- Write simple, clear scripts that you and others can read later
- Save raw data (as read only!)
- Publicly release raw data and scripts
- Start with a good experimental design. This means asking for help from a statistician BEFORE running any samples.

Operating Systems

Linux vs Unix

- UNIX
  - Operating system developed in the 1970s at Bell Labs
  - Copyrighted name
  - Usually costs a lot of money (can be free for certain types of development)
  - Common for large corporate computers (servers and mainframes)

Linux vs Unix



Linux

- Linus Torvalds was frustrated that UNIX required a license
- So he wrote his own operating system from scratch that mimics UNIX
- released in 1991
- Free to use, open source
- One of the most prominent and important free, open source software projects

PS – Linus Torvalds also wrote git. So he basically revolutionized open source computing. Twice.

Because it is open, it has been ported and cloned, yielding many “flavors” or “distros”, all slightly different

(To further complicate life, UNIX itself has clones and ports that aren’t Linux)

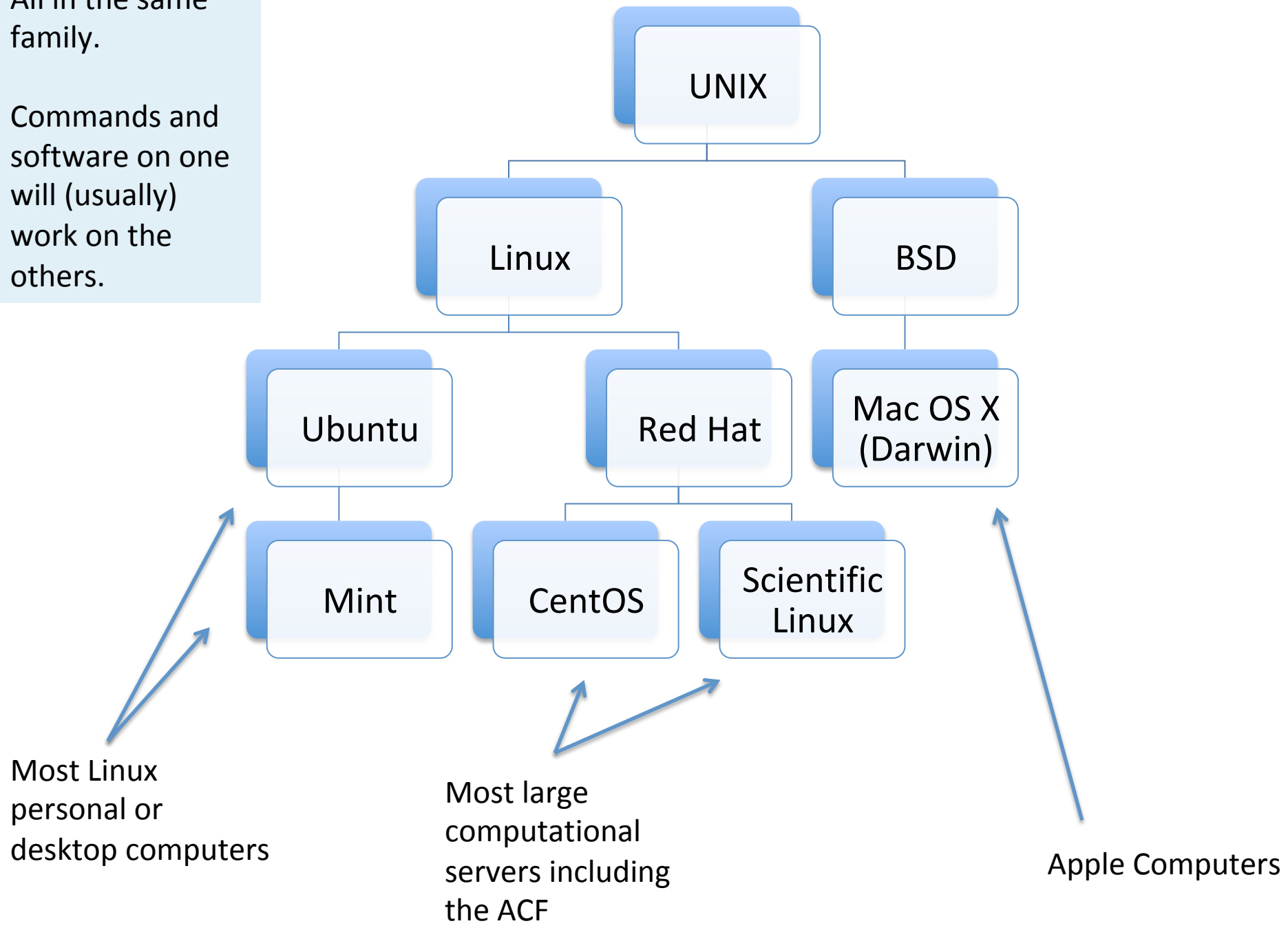
Apple?

- Mac OS X uses their own version of UNIX
- Called Darwin
- The Darwin part of OS X is open access
- The GUI part is not
- Partially a port of Berkeley Software Distribution (BSD)
- Certified as a UNIX OS

Darwin mascot is Hexley the Platypus

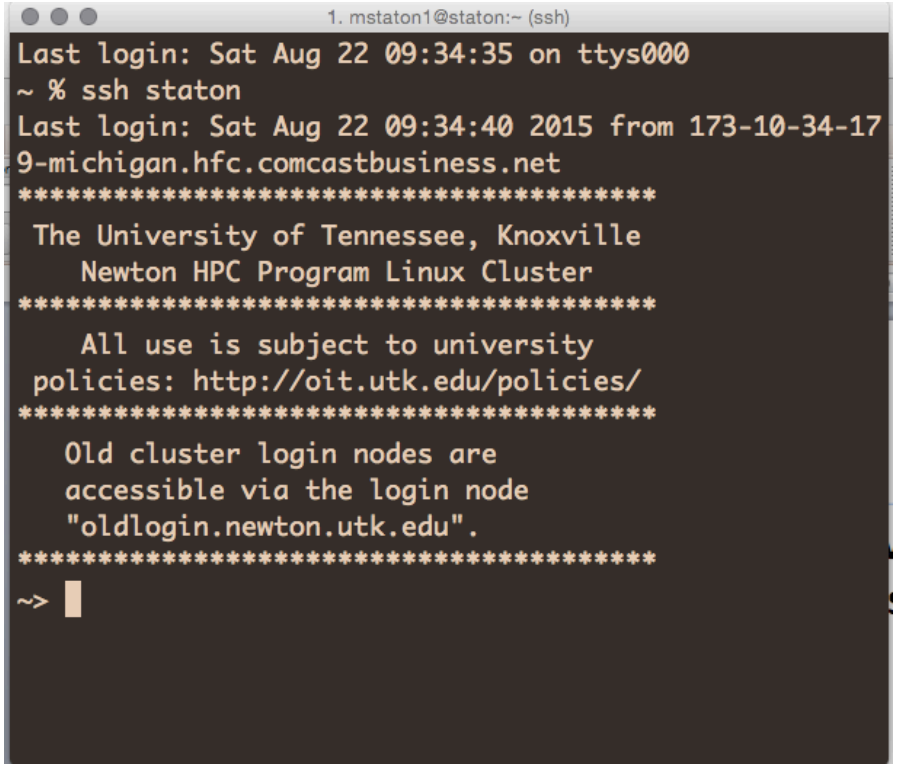
All in the same family.

Commands and software on one will (usually) work on the others.



Shell

- A shell is any user interface allowing access to the functions of an operating system
  - Command line (CLI)
  - Graphical (GUI)
- Most often “shell” or “terminal” refers to a CLI



Shell Variety – sh and bash

Variety of shell types:

- sh or Bourne Shell
  - the original shell still used on UNIX systems and in UNIX-related environments
  - this is the basic shell, a small program with few features.
  - No longer standard shell, but still available on every Linux system for compatibility reasons.

- bash or Bourne Again shell
  - the standard GNU shell
  - a set of add-ons and plug-ins to the bourne shell
  - Good for beginners, loved by many pros
  - This is our shell

- Others: csh or C shell, tcsh or turboC shell

But why?

- Speed, Power, Flexibility
- Can customize commands and run software with a few key strokes
- Can send information between programs with pipes
- Can operate on hundreds to millions of files
- Can have many different jobs running simultaneously across many computers
- Easier to write code and release software – which means you can use the best software from open source researchers
- Modular workflows and components
  - We can experiment with different pieces of software at each stage of analysis (or substitute in our own!)
  - Reuse
  - Examine results at each stage of analysis

Geeks and repetitive tasks