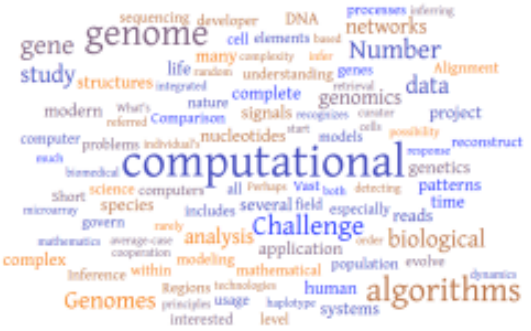
Computational Biology
BIO 253/453



Justin Fay

TA: Emiliano Marti

TA: Faye Romero

Today's objectives

- What is computational biology?
- Course objectives, what you will learn
- Course format, grading, lab and lecture materials

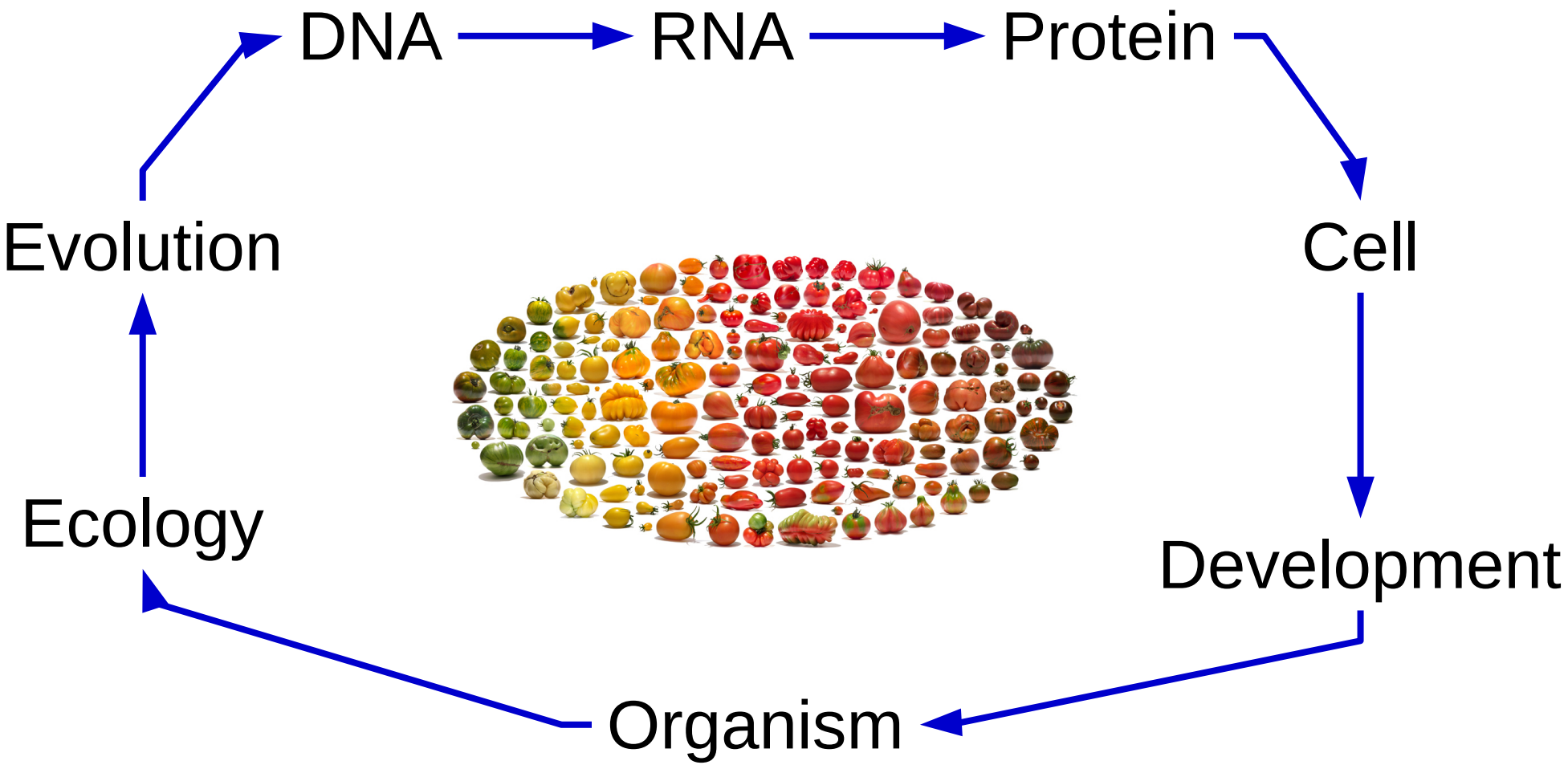
What is computational biology?

Computational biology – the study of biology using computational techniques. Involves the development and application of data-analytical and theoretical methods, mathematical modeling and computational simulation techniques to the study of biology.

Bioinformatics – (applied) the creation of tools (algorithms, databases) that solve biological problems. Involves the development of methods and software tools to understand biological data.

Model/theory vs software (e.g. BLAST)

A central problem in biology: Decoding the blueprint for life



DNA (genomes) are large and require computational analysis
Genomics era: from single gene studies to genomics (requires compbio)

This course: Computational Genomics (RNA+DNA)

- Sequence/Assembly
- Genes
- Regulation
- Model organisms
  - Alignment
  - Evolution
  - Phylogenetics
  - Comparative genomics

Genome – Organism

Sequencing v. decoding

Model organism discoveries

Nobel Prize	Year	Recipients	Organism
Cellular origin of retroviral oncogenes	1989	J. Michael Bishop, Harold E. Varmus	Chicken
Regulators of the cell cycle	2001	Leland Hartwell, Timothy Hunt, Paul Nurse	Yeast, sea urchins
Genetic regulation of development and programmed cell death	2002	John Sulston, Robert Horvitz, Sydney Brenner	Nematode worm
RNA interference	2006	Andrew Fire, Craig Mello	Nematode worm
Chromosome protection by telomeres and telomerase	2009	Elizabeth H. Blackburn, Carol W. Greider and Jack W. Szostak	Frog, Mouse, Tetrahymena
The development of in vitro fertilization	2010	Robert G. Edwards	Mouse
Mature cells can be reprogrammed to become pluripotent	2012	John Gurdon, Shinya Yamanaka	Frog, Mouse
The circadian rhythm	2017	Jeffrey Hall, Michael Rosbash, and Michael Young	Drosophila (fruit fly)

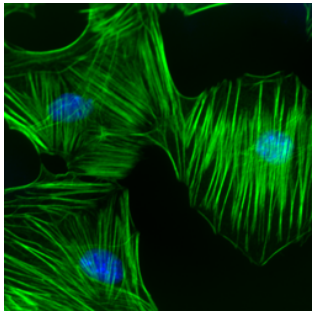
Other areas of computational biology: biophysics + imaging

Genomics

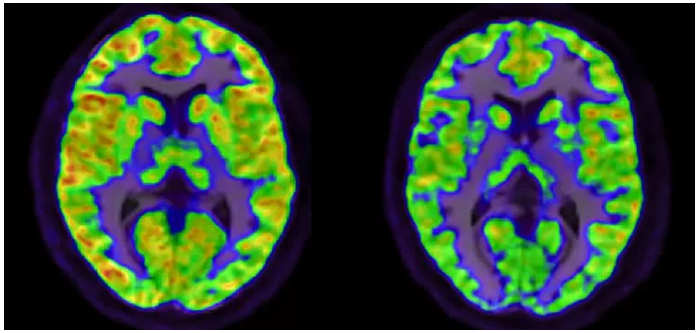
```
GCACGATCGATCA
GCACGAACCATCA
GCA-GATCCATCA
GCA-GATCGATCA
```

This course, but Models/Methods are general

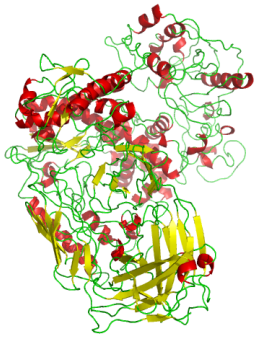
Cell biology & Development



Neuroscience



Biochemistry

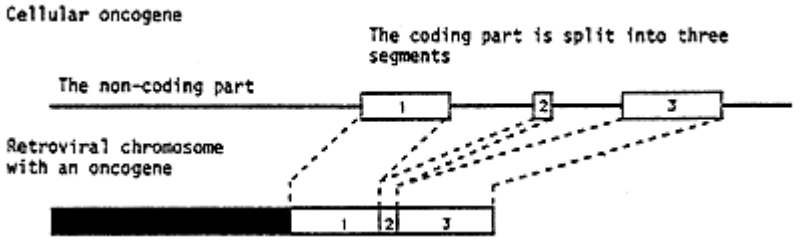


Proto-oncogenes example

barred Plymouth Rock hen

- Peyton Rous showed that a tumor from a barred Plymouth Rock hen could be transplantable to chickens of the same breed (Rous, 1910)
- The oncogene v-src (Rous Sarcoma Virus): a sarcoma causing virus from chicken is derived from c-src a non-receptor tyrosine kinase proto-oncogene
- oncogenes (cancer-causing) are activated (mutation/expression) from proto-oncogenes, found in many organisms including humans
- Bishop and Varmus were awarded the Nobel Prize in Physiology or Medicine in 1989 for their discovery of the cellular origin of retroviral oncogenes. (How: computational biology)
- c-src is a tyrosine kinase protein that promotes survival, angiogenesis, proliferation and invasion pathways (v-src constitutively active form of c-src)

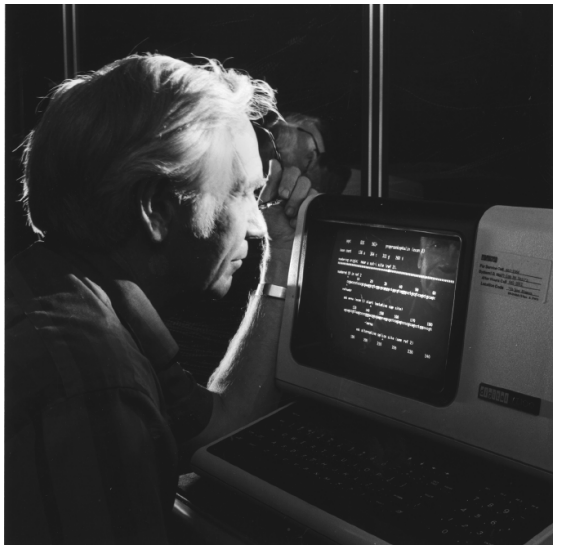
Sequence similarity between v-src and c-src



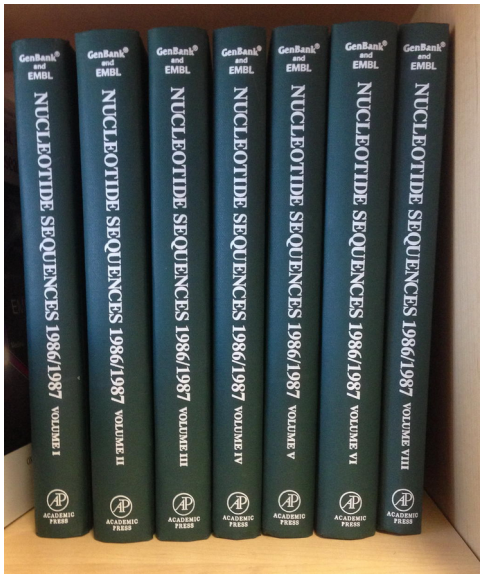
Value of GenBank

GenBank – the first database of nucleotide and protein sequences

Walter Goad and colleagues established the Los Alamos Sequence Database (1979), which, in 1982 became the public GenBank



Walter Goad

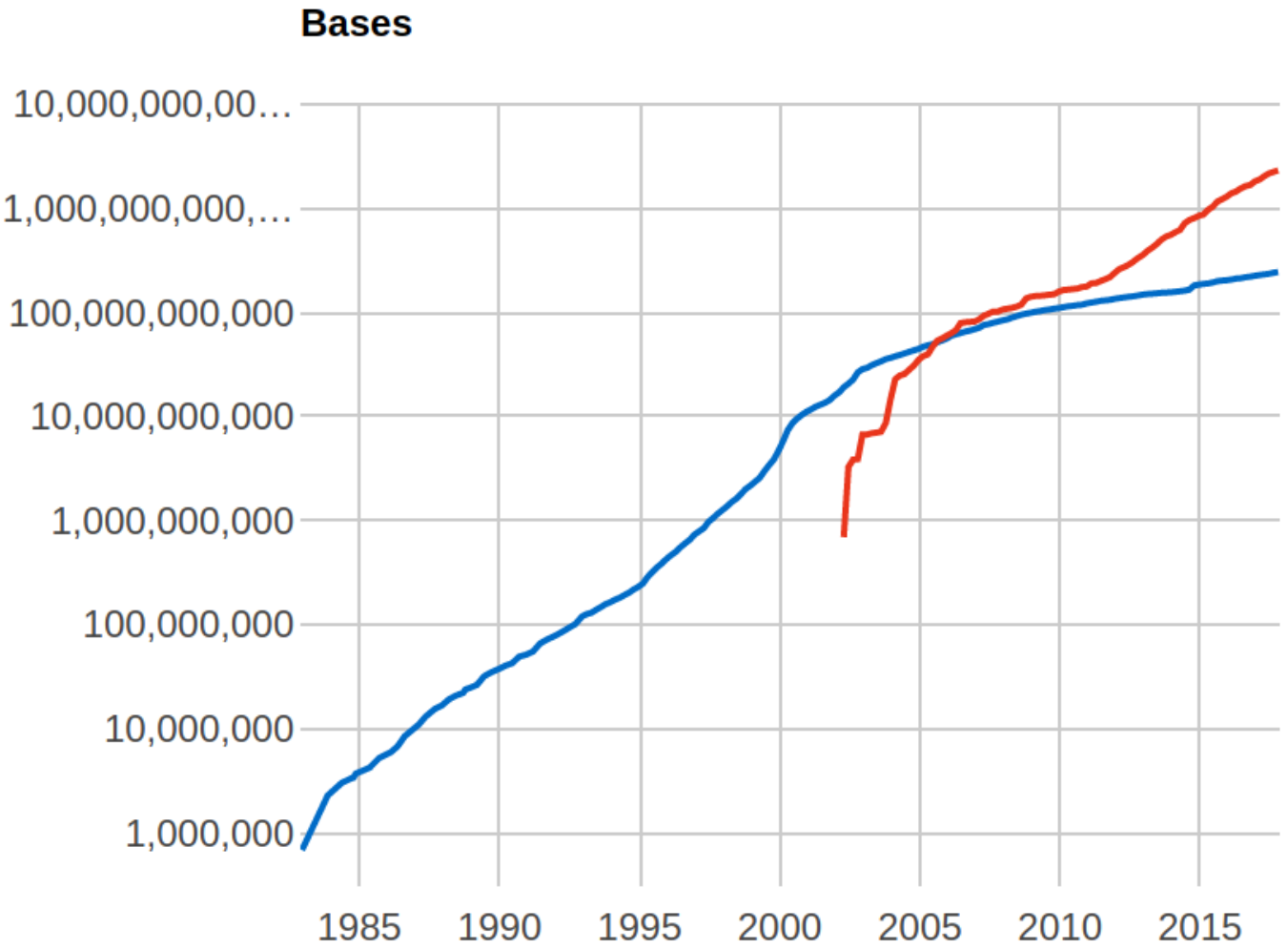


GenBank and EMBL: Nucleotide Sequences 1986/1987 Volumes I to VII.

David Lipman and colleagues developed methods (BLAST, 1990) to allow easy, web-based access to sequence databases

David Lipman

Growth of sequence data in GenBank at the National Center for Biotechnology Information (NCBI)

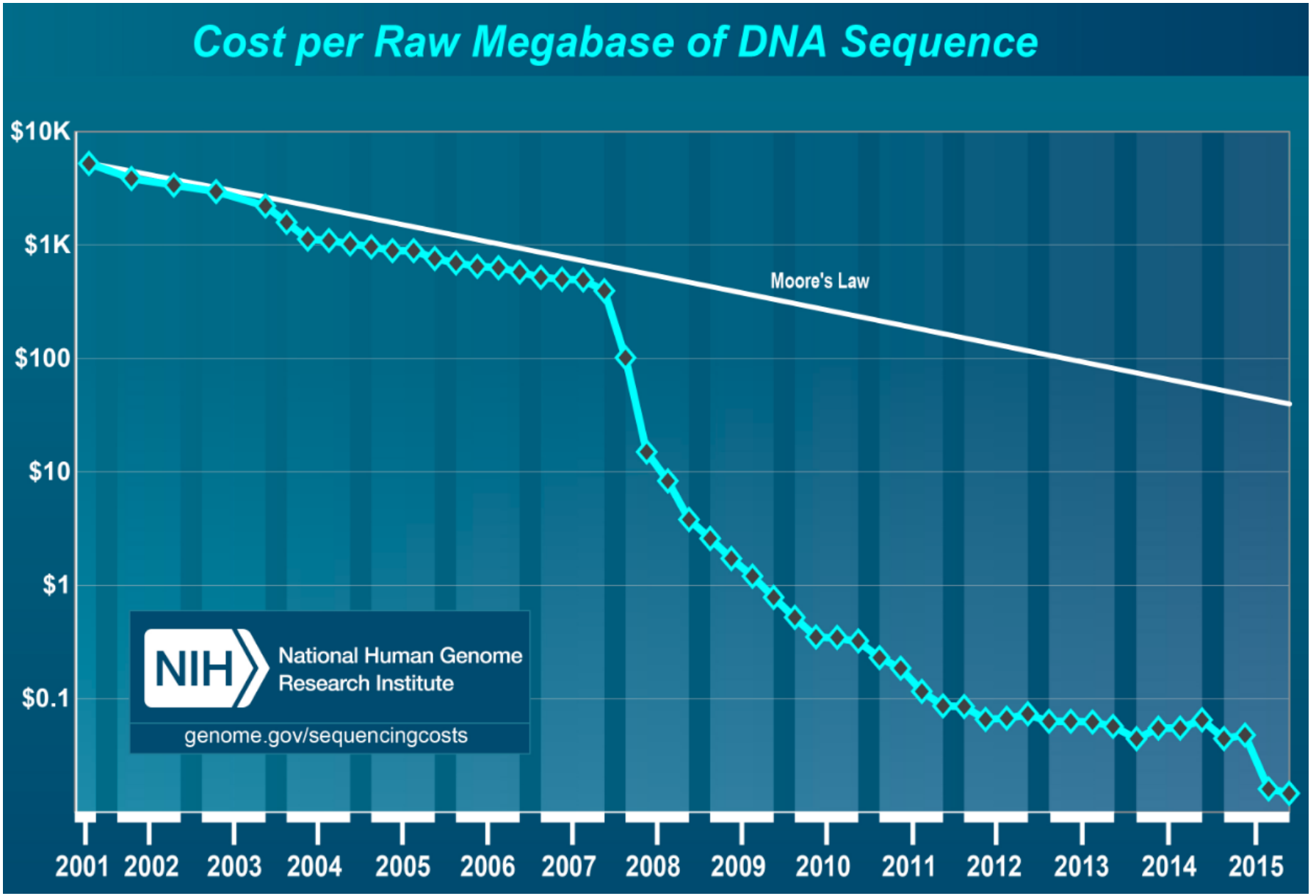


Red = whole genome sequences (WGS)
Blue = other sequences

https://www.ncbi.nlm.nih.gov/genbank/statistics/

Cost of sequencing

Moore's law = doubles every 18-24 months



Genomics is Big Data

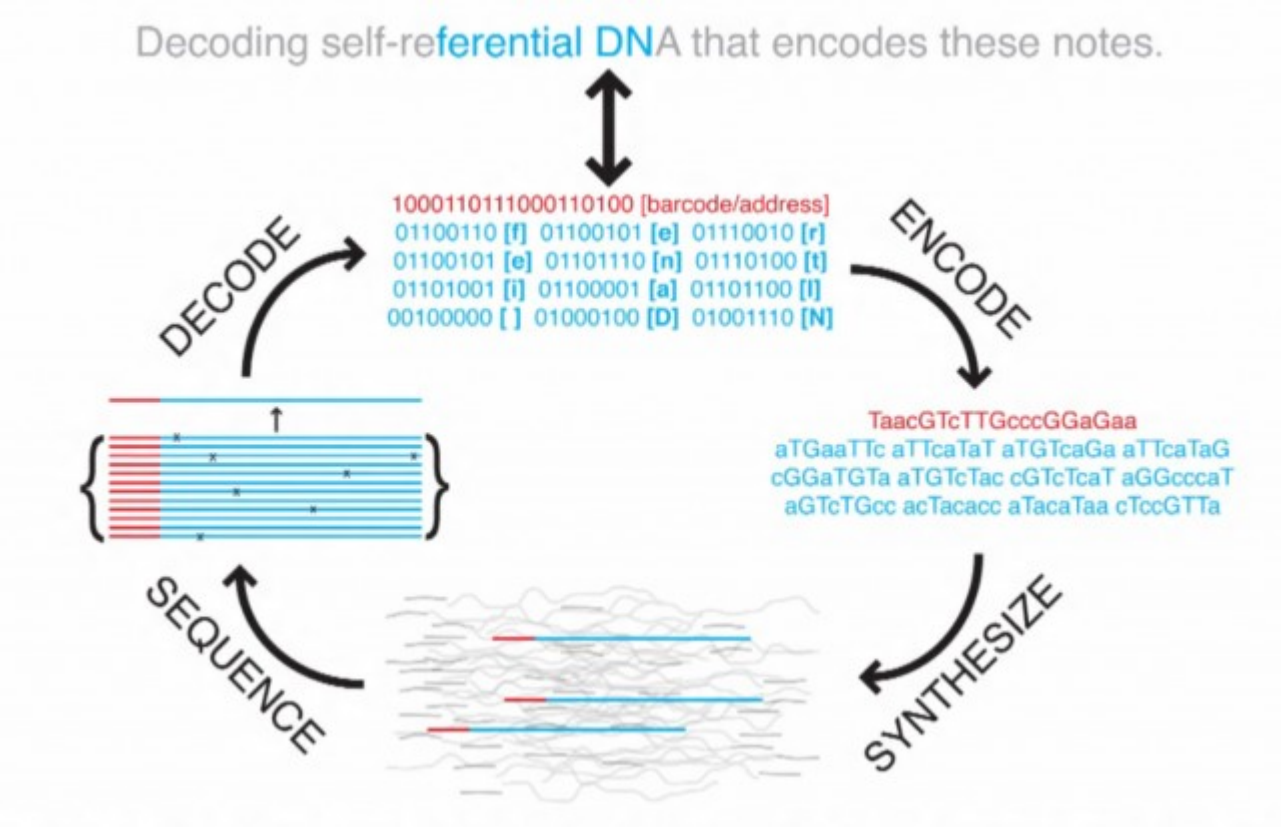
Data Phase	Astronomy	Twitter	YouTube	Genomics
Acquisition	25 zetta-bytes/year	0.5–15 billion tweets/year	500–900 million hours/year	1 zetta-bases/year
Storage	1 EB/year	1–17 PB/year	1–2 EB/year	2–40 EB/year
Analysis	In situ data reduction	Topic and sentiment mining	Limited requirements	Heterogeneous data and analysis
	Real-time processing	Metadata analysis		Variant calling, ~2 trillion central processing unit (CPU) hours
	Massive volumes			All-pairs genome alignments, ~10,000 trillion CPU hours
Distribution	Dedicated lines from antennae to server (600 TB/s)	Small units of distribution	Major component of modern user's bandwidth (10 MB/s)	Many small (10 MB/s) and fewer massive (10 TB/s) data movement

doi:10.1371/journal.pbio.1002195.t001

Decimal		
Value	Metric	
1000	kB	kilobyte
1000^2	MB	megabyte
1000^3	GB	gigabyte
1000^4	TB	terabyte
1000^5	PB	petabyte
1000^6	EB	exabyte
1000^7	ZB	**zettabyte**
1000^8	YB	yottabyte

Big Data: Astronomical or Genomical?

DNA as a storage device



One gram of DNA can store 700 terabytes of data

$7000 to synthesize 2 megabytes of data and another $2000 to read it

Algorithms and models used in computational biology

A *model* is a parametric explanation of the observations of interest, while an *algorithm* is a set of instructions for solving a problem, e.g. inferring the optimal value of a model's parameter.

Probabilistic Models/Methods

- maximum likelihood
- Bayesian
- machine learning
- Markov chain models
- hidden Markov models
- Markov chain Monte Carlo (MCMC)

Algorithms/Methods

- Sequence (string) algorithms
  - sort/search/substring algorithms
- Optimization algorithms
  - linear programming
  - dynamic programming
  - greedy algorithms
  - heuristic methods
- Expectation-maximization (EM)

Generality of algorithms and methods

CGGCGTTATGACGTA

Text → Language

Genes → Organism

Automated speech recognition

- Dynamic time warping (1960s)
- Hidden Markov models (HMMs, 1980s)
- Neural networks (late 80s)
- Deep feedforward neural networks (DNN, by 2010)

Gene finding

- Pattern (string) finding (1980s)
- Hidden Markov models (1990s)

Gene regulation

- Regulatory motifs (strings) (1980s)
- HMMs + others (1990s)
- Deep feedforward neural networks (DNN, 2010s)

Driven by advances in computing power

Science is based on models

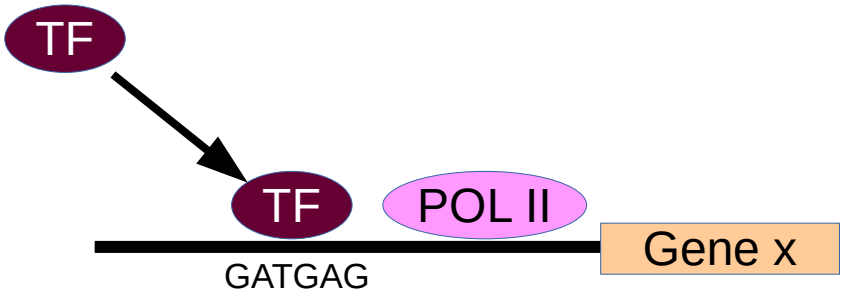
All science is based on models, and every scientific model comprises three distinct stages: statement of well-defined hypotheses; deduction of all the consequences of these hypotheses, and nothing but these consequences; confrontation of these consequences with observed data.

-Maurice Allais

All models are wrong, but some models are useful.

-George Box

Computational Biology uses mathematical/statistical models



Biology: GATGAG is necessary and sufficient to induce expression

Computational biology:
P(TF binding | sequence) or
P(expression | [TF] & sequence)

Computational Biology uses mathematical/statistical models

Model evaluation (algorithm/computer)

- Ability to explain past observations
- Ability to predict future observations
- Refutability, enabling estimation of the degree of confidence in the model
- Simplicity, or even aesthetic appeal (Occams' Razor)
- Cost of use, especially in combination with other models

Algorithms

- Roughly speaking, an algorithm is a sequence of instructions that one must perform in order to solve a well-formulated problem. (Pevzner)

- You can use a pseudocode to write an algorithm, and a programming language to implement in.

Example of a pseudocode
MAX(a,b) below computes the maximum of two numbers:

```
MAX(a,b)
1.  if a < b
2.          return b
3.  else
4.          return a
```

A simple example

Model of a gene

A gene is a sequence of nucleotides that encodes a protein sequence between 20 and 2000 residues in length. A gene starts with methionine and ends with a stop codon.

Gene finding algorithm

Search for and identify all sequences that: start with ATG, end with either TAG/TAA/TGA, and is between 60 and 6000 nucleotides in length.

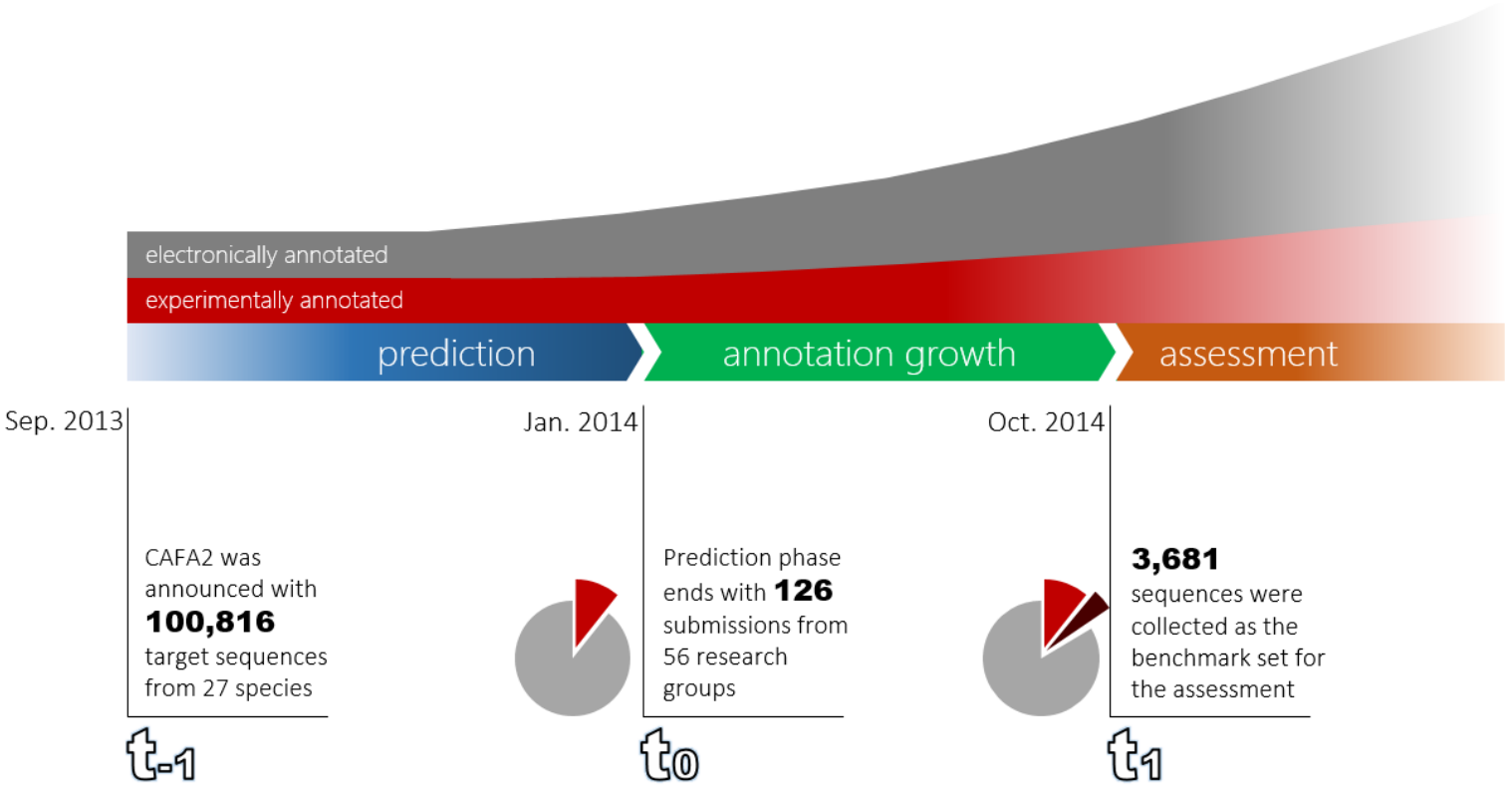
Benchmarking

Benchmarking is running a program in order to assess its relative performance, normally through comparison with standards or other programs.

- Performance can be measured in terms of time or memory required to solve a problem.
- Performance can also be measured by how often a program returns a correct result.
- Simulations and datasets with known answers are two ways in which algorithms are commonly benchmarked.

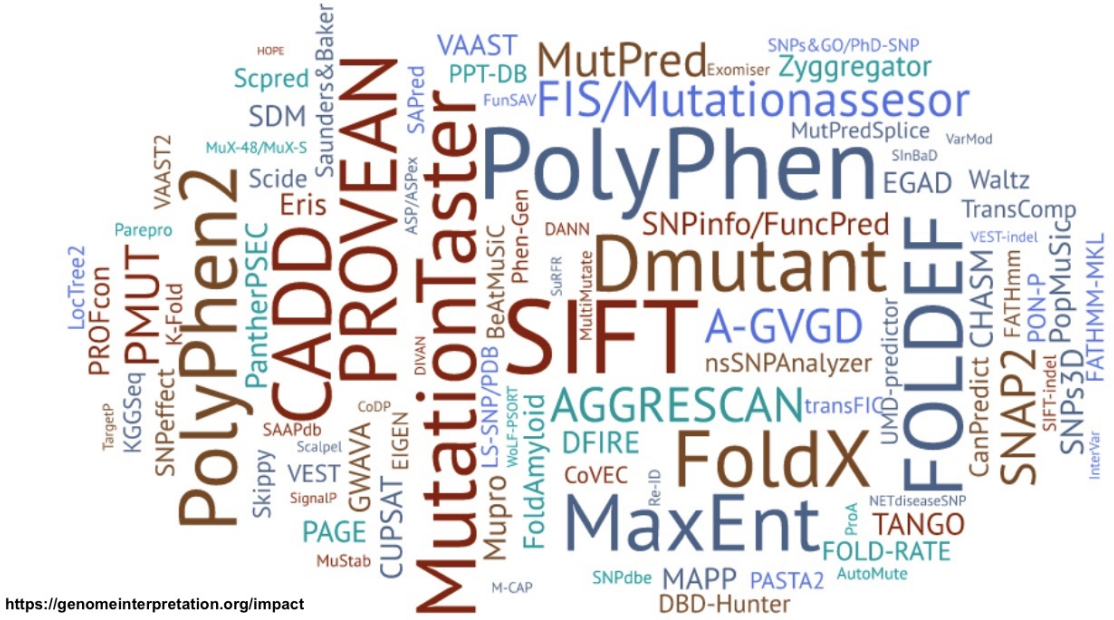
Computational Biology Grand Challenges

CAFA: There are many proteins in the databases for which the sequence is known, but the function is not. Critical Assessment of protein Function Annotation algorithms (CAFA) is an experiment designed to provide a large-scale assessment of computational methods dedicated to predicting protein function.



Predicting disease mutations

CAGI: The Critical Assessment of Genome Interpretation (CAGI) is a community experiment to objectively assess computational methods for predicting phenotypic impacts of genomic variation and to inform future research directions.



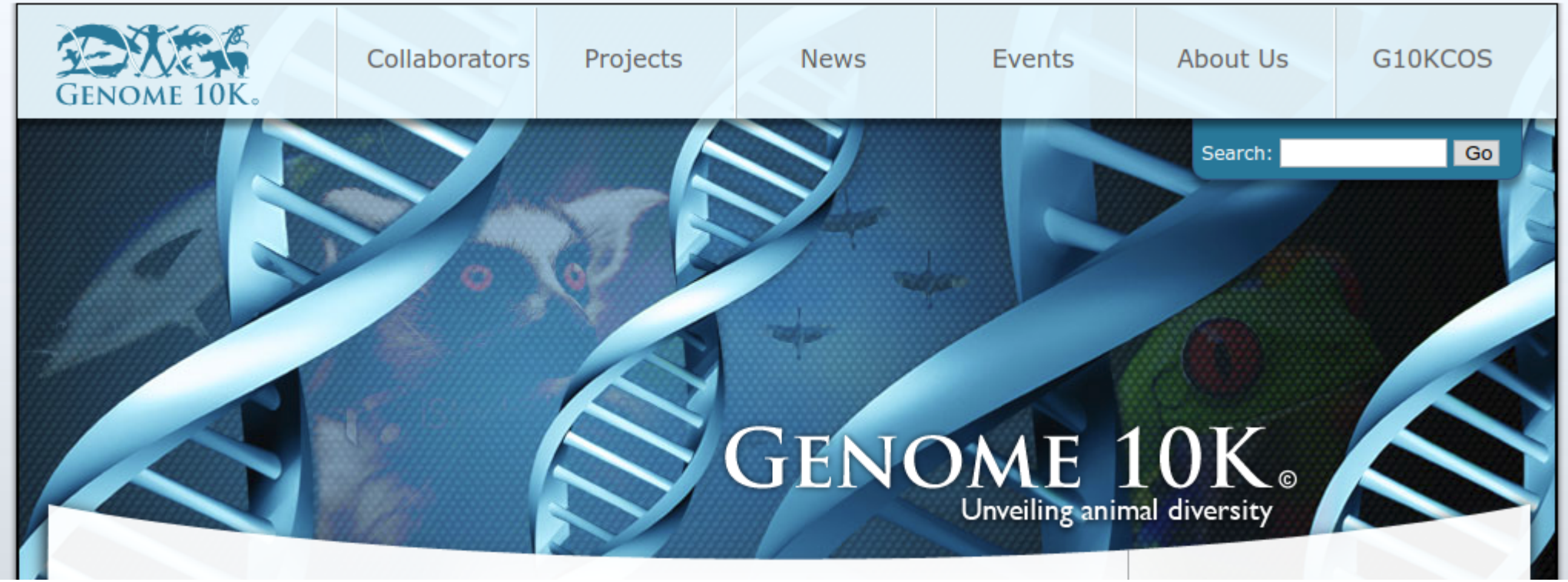
Assemblathon

The Assemblathon is a set of periodic collaborative efforts that all help improve methods of genome assembly.

The overall goal of each Assemblathon event is to have participating groups try to use their own software to each assemble one or more genomes that the organizers of the Assemblathon will make available

Alignathon

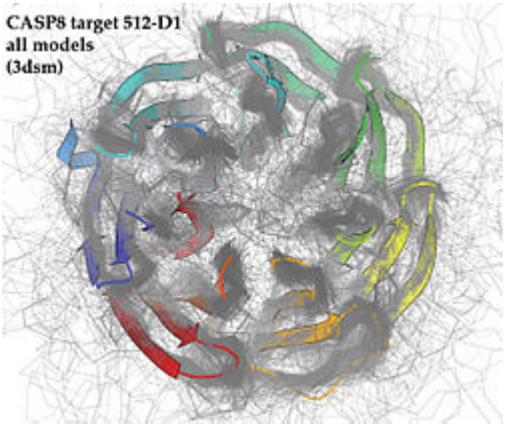
The Alignathon is a collaborative project to assess whole genome aligners and promote development of the field of whole genome alignment. We present here three sets of simulated genomes created using the Evolver genome evolution simulation suite (controlled by evolverSimControl) and one real data set comprised of the 20 fly genomes.



Critical Assessment of protein Structure Prediction (CASP)

Our goal is to help advance the methods of identifying protein structure from sequence. The Center has been organized to provide the means of objective testing of these methods via the process of blind prediction.

The Critical Assessment of protein Structure Prediction (CASP) was started in 1994 and involves double blind experiments. Neither predictors nor the organizers and assessors know the structures of the target proteins at the time when predictions are made.



Course objectives

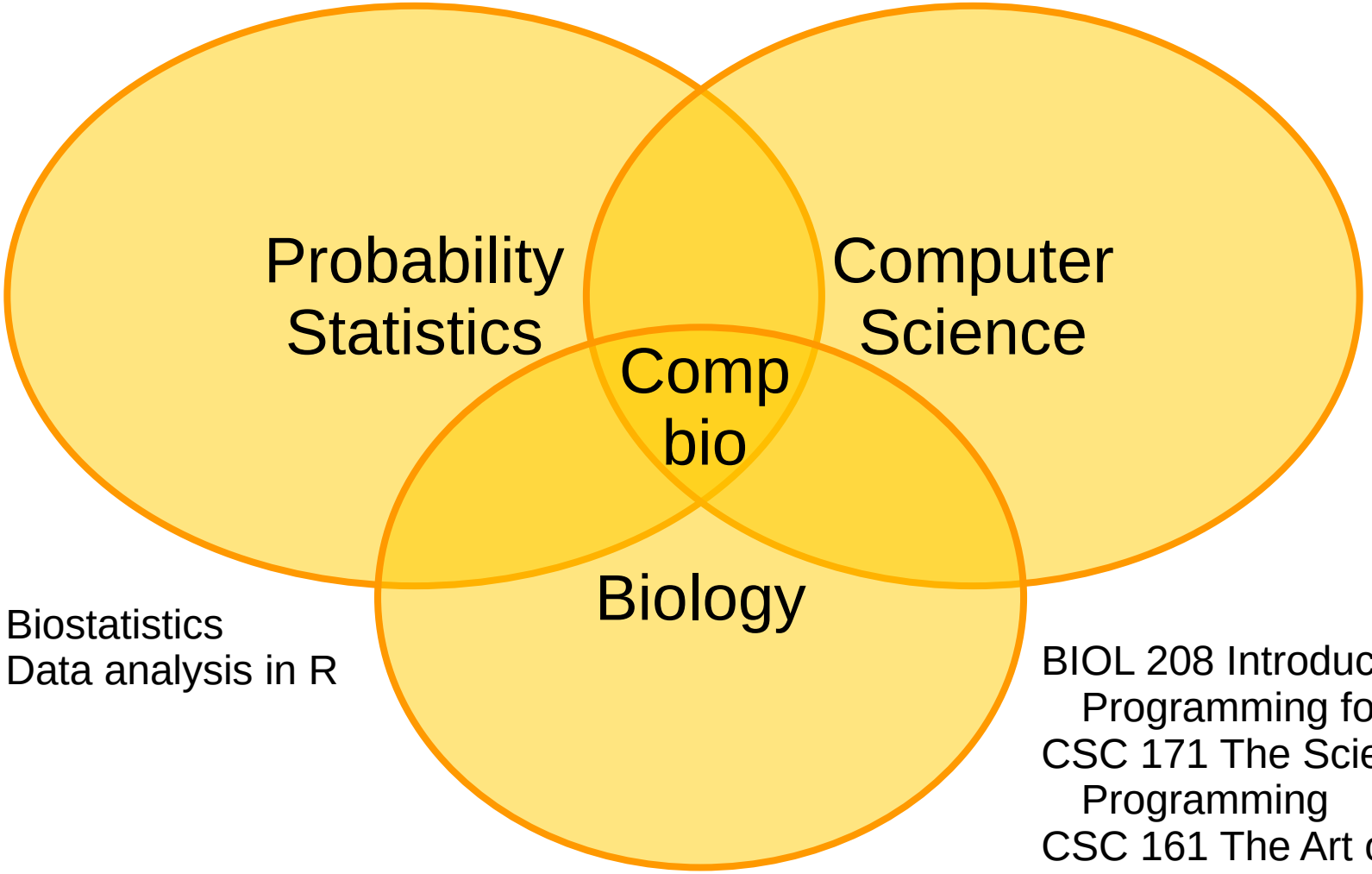
1) Models: understand how probabilistic models work and what their utility is an answering biological problems.

2) Algorithms: understand how to encode and apply computational algorithms to biology.

3) Genomics: understand current challenges and solutions used in biological sequence analysis.

4) Lab: learn how to code and apply models and algorithms in practice.

Background and Prerequisites



BIOL 214 Biostatistics
BIOL 218 Data analysis in R

BIO 257 Applied Genomics with Lab
BIO 259 Applied Population Biology

BIOL 208 Introduction to Programming for Biologists
CSC 171 The Science of Programming
CSC 161 The Art of Programming
BIO 152 Introduction to UNIX
BIO 153 Bootcamp in Python

	Topic	Lab
Week 1	**Course introduction**	
Jan 12	Introduction to computational biology	
Week 2	**Algorithms**	
Jan 19	History, algorithms and challenges	Lab01 Linux and Bash
Week 3	**Strings**	
Jan 24	BW transform, string search, string analysis	Lab02 Intro to Python
Week 4	**Sequence alignment**	
Jan 31	Dynamic programming, BLAST	Lab03 String Search
Week 5	**Molecular evolution**	
Feb 7	Substitution models, Markov models, likelihood	Lab04 Markov Models
Week 6	**Phylogenetics**	
Feb 14	Trees, pruning algorithm, MCMC, selection	Lab05 Ancestral states
Week 7	**Comparative genomics**	
Feb 21	Genome alignment, mutation, disease, ROC	Lab06 Simulation
Week 8	**Metagenomics**	
Feb 28	Midterm exam in class (3/2)	
	Spring break	
Week 9	**Genome annotation and function**	
Mar 14	Hidden Markov models	Lab07 HMMs
Week 10	**Motifs**	
Mar 21	PWMs search and comparison, motif finding	Lab08 EM and motifs
Week 11	**Machine learning**	
Mar 28	Clustering, classification, machine learning	Lab09 Clustering
Week 12	**Bioinformatics**	
Apr 4	Bioinformatic pipelines and databases	Lab10 Bioinformatics
Week 13	**Biological models**	
Apr 11	Model development, selection, deep learning	
Week 14	**Current topics**	
Apr 18	TBA	
Week 15	**Recitation**	
Apr 25	Final exam in class (4/27)	

Lecture/Lab format

Lectures

- Slides on blackboard
- Class exercises (questions)

Labs

- Lab assignments on blackboard
  - Python using Jupyter Notebooks
  - Linux and Bash (Lab 01)
  - Copy lab materials to your home folder on bluehive

Programming languages

C : a general purpose language with lasting use in many important applications, ranging from operating systems to application software

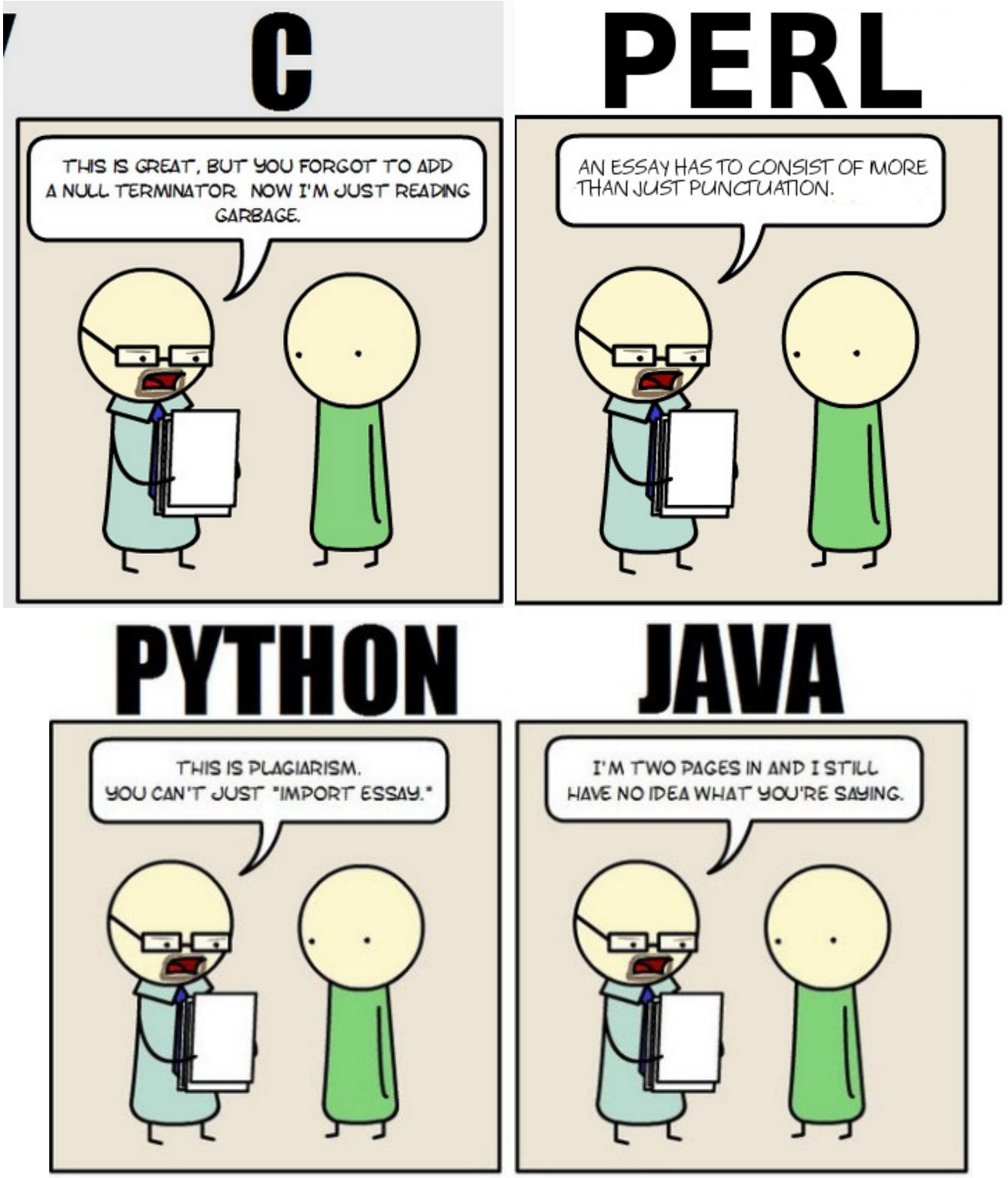
Java: a general purpose *object oriented* language specifically designed to have as few *dependencies* as possible.

Perl: a family of *high level*, general purpose dynamic programming language that borrows features from C, shell script, awk and sed. Used for 'scripting', graphics programming, sys administration, and as a 'duct tape' language.

Python: *high level*, general purpose dynamic programming language whose design philosophy emphasizes code readability, and allows for programming in fewer lines of code.

Bash: an shell-compatible command language interpreter that executes commands read from the standard input or from a files.

- Compiled program - written in human-readable form are translated (compiled) into a language that microprocessors can understand directly.
- Interpreted/scripting language - stored as text files and processed by an interpreter every time they are run, adding to computational overhead.


C
THIS IS GREAT, BUT YOU FORGOT TO ADD A NULL TERMINATOR. NOW I'M JUST READING GARBAGE.
PERL
AN ESSAY HAS TO CONSIST OF MORE THAN JUST PUNCTUATION.
PYTHON
THIS IS PLAGIARISM. YOU CAN'T JUST "IMPORT ESSAY."
JAVA
I'M TWO PAGES IN AND I STILL HAVE NO IDEA WHAT YOU'RE SAYING.

Grading

	Points	Percentage of final grade
Lab assignments	200	67
Midterm	50	17
Final	50	17

10 lab assignments, 20 points each

Lab bonus points: up to 20/20

Late assignments: You have a total of 5 days of late credit that you can use throughout the semester without penalty. For example, if a lab assignment is due Tuesday, you will use 1 credit if you turn it in up to the due time on Wednesday, 2 credits for the same time on Thursday, etc. If all of your credit is used then you will receive half the number of points earned on the assignment. For example, a lab assignment graded 18/20 will yield 9 points.

Academic Honesty

All assignments and activities associated with this course must be performed in accordance with the University of Rochester's Academic Honesty Policy: www.rochester.edu/college/honesty. The instructor takes violations of academic honesty seriously and is required to document and act on any evidence of improper academic conduct. Violations including plagiarism, misuse of user computer accounts, and cheating.

Computer lab assignments: Computer lab assignments including code and answers to questions are to be completed by each student. You are encouraged to use publicly available online information, e.g. wikipedia, stackoverflow, as well discuss problems with students currently enrolled in the the course. You may not solicit or obtain help from others, particularly previously enrolled students, via online message board posts or other solicitations. Sharing of data electronically or otherwise is not allowed. *While you may discuss problems with others, you may not share data, code or answer electronically or otherwise.*

How to get an “A”

Class exercises

- complete all, ask questions in class

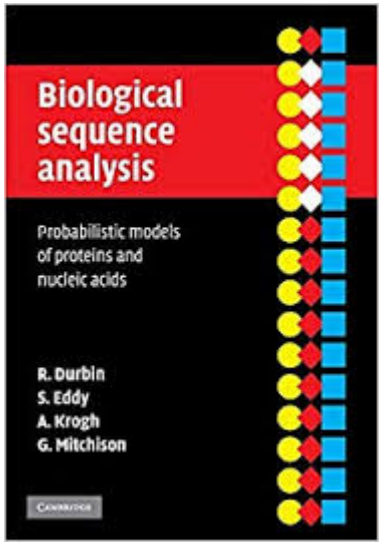
Midterm/Final

- review all in class exercises

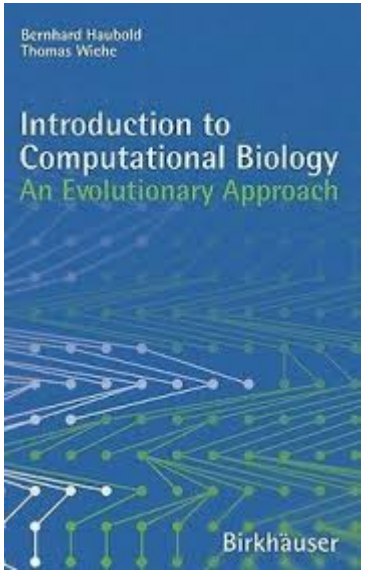
Lab assignments

- comment all code
- TA office hours
- resources
  - students, TAs and instructor
  - google searches, stackoverflow, wikipedia

Reserve Material (Carlson)

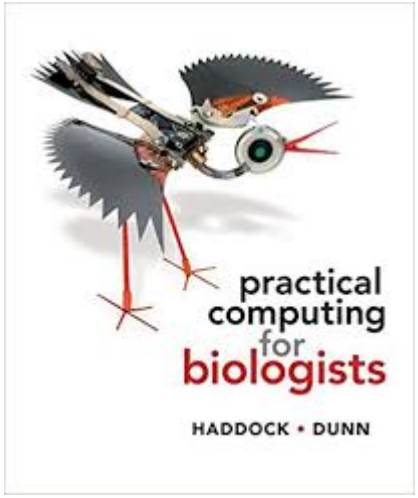


Durbin R, Eddy SR, Krogh A, Mitchison G. Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge university press; 1998 Apr 23.

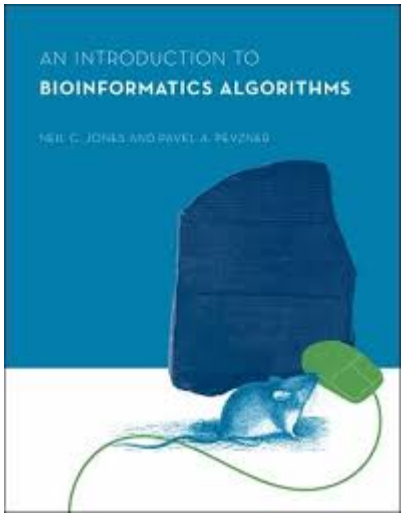


Haubold B, Wiehe T. Introduction to computational biology: an evolutionary approach. Springer Science & Business Media; 2006 Aug 9.

Haddock SH, Dunn CW. Practical computing for biologists. Sunderland, MA, USA: Sinauer Associates; 2011.



Jones NC, Pevzner PA, Pevzner P. An introduction to bioinformatics algorithms. MIT press; 2004.



Course Information

- Blackboard
  - Academic honesty
  - Syllabus
  - Lecture/labs

While you may discuss problems with others, you may not share data, code or your answer electronically or otherwise.

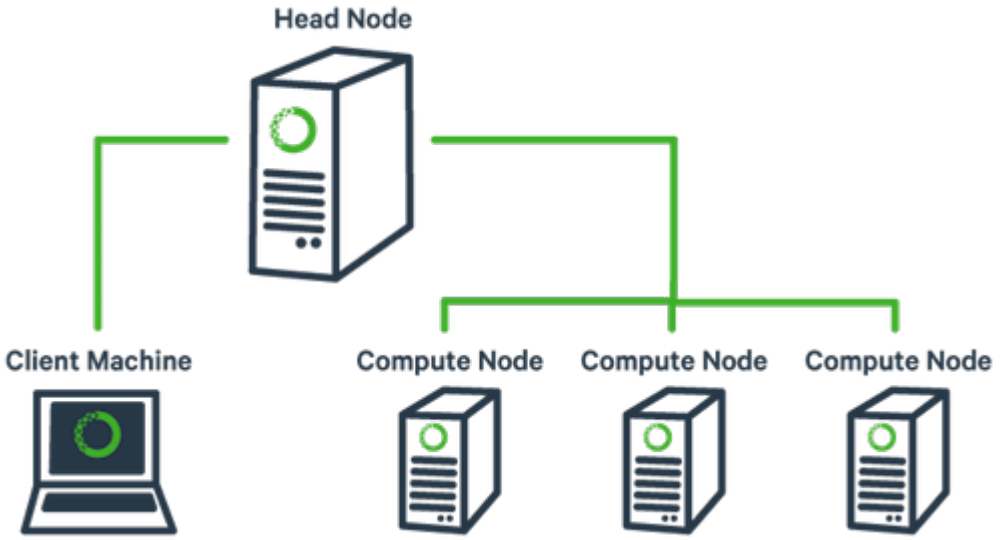
Help debugging is ok, as long as you act like a TA.

- Computer labs (Jupyter notebooks)
  - bluehive.circ.rochester.edu is the cluster
  - https://jupyter.circ.rochester.edu is the link for Jupyter lab notebook
  - PC, Jupyter and Python install using Anaconda

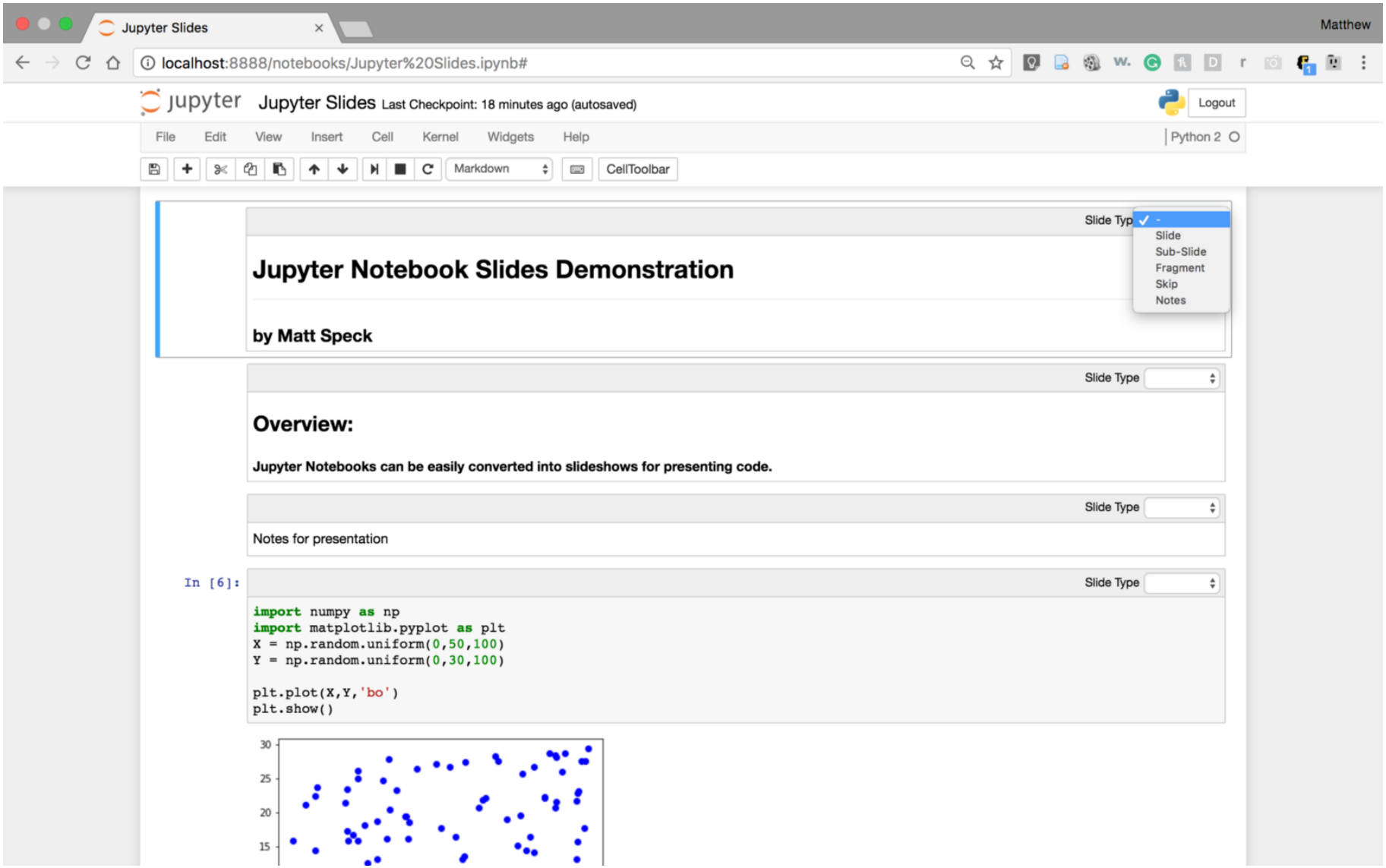
Center for Integrated Research Computing (CIRC)

The BlueHive cluster is CIRC's primary Linux cluster for demanding computations.

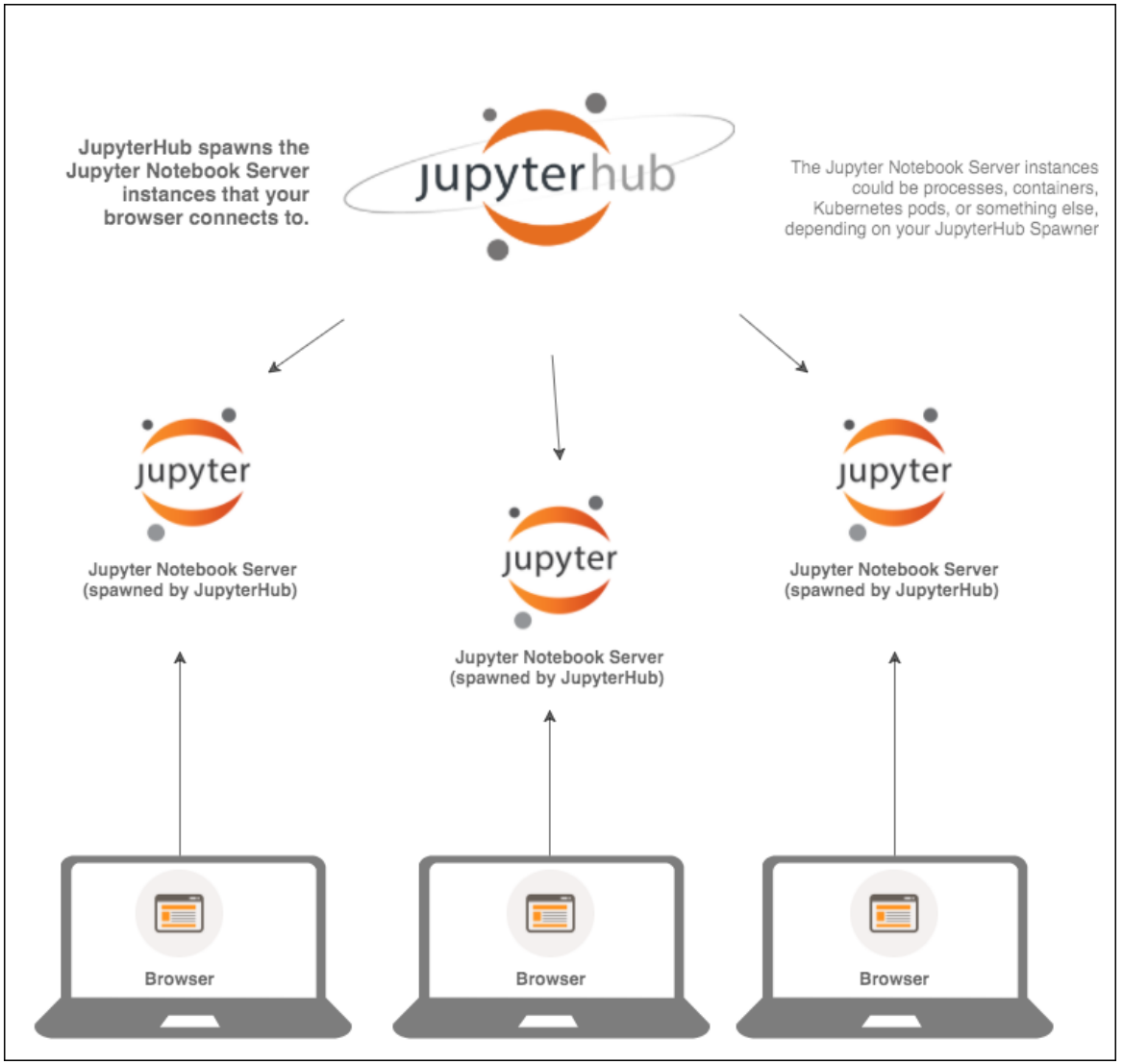
- 284 nodes
- 5,656 CPU cores
- 21 TB RAM
- 210 TeraFLOPS (floating point operations per second)



Jupiter notebooks



Jupyter Notebook online



JupyterHub is installed on bluehive:

https://jupyter.circ.rochester.edu

Install Jupyter notebook/lab on your computer using:

https://www.anaconda.com/

Lab 01 | Linux and Bash

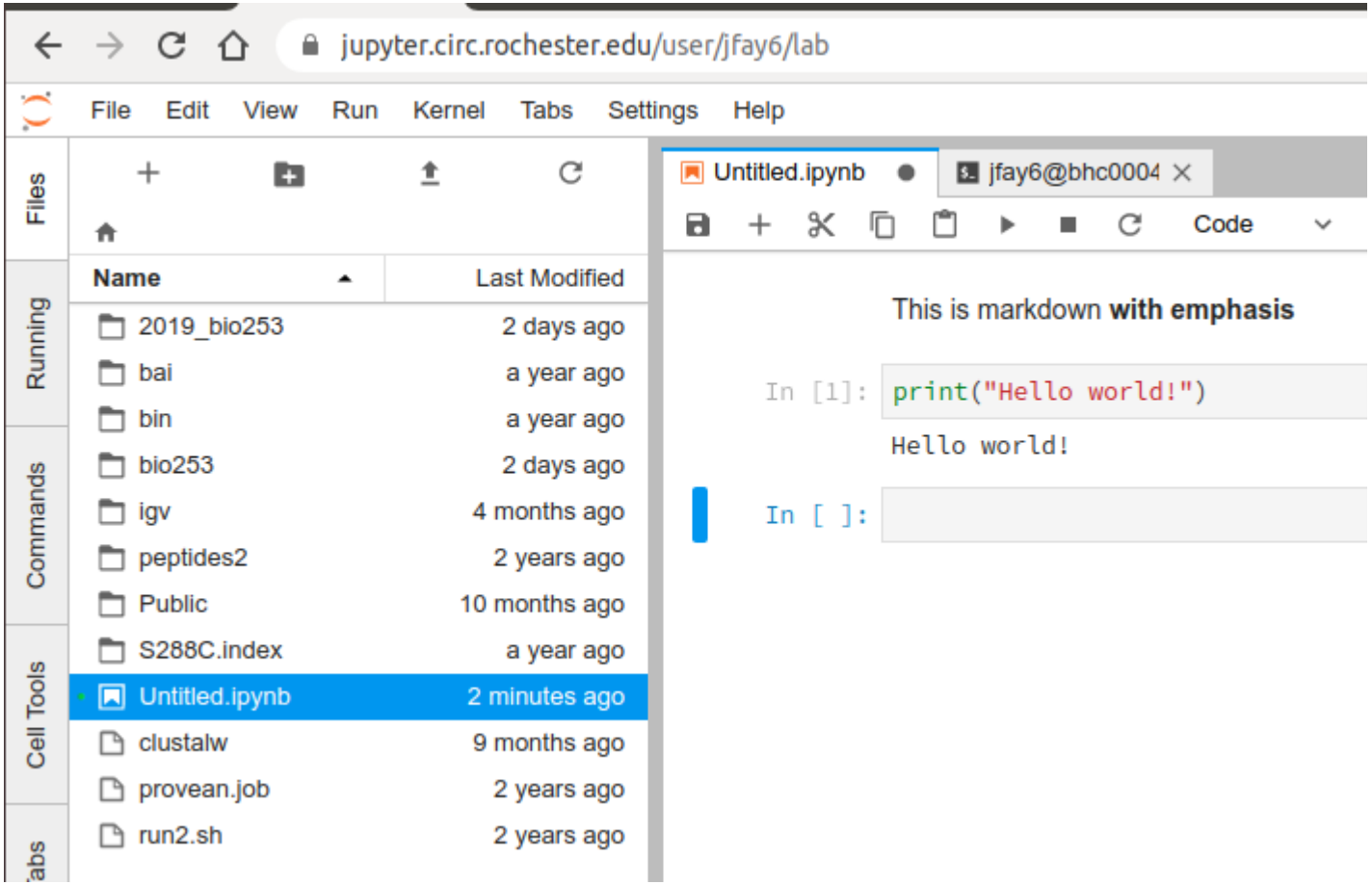
Before Lab on Tuesday/Wednesday

- Read Prelab
- Login to Jupyter on bluehive (2 factor authentication) copy lab to your home directory OR
- Install Jupyter Lab on your PC using Anaconda (10 minutes or more)
  - Download lab assignments from blackboard or bluehive (ssh/scp/sftp)

Lab on Tuesday/Wednesday

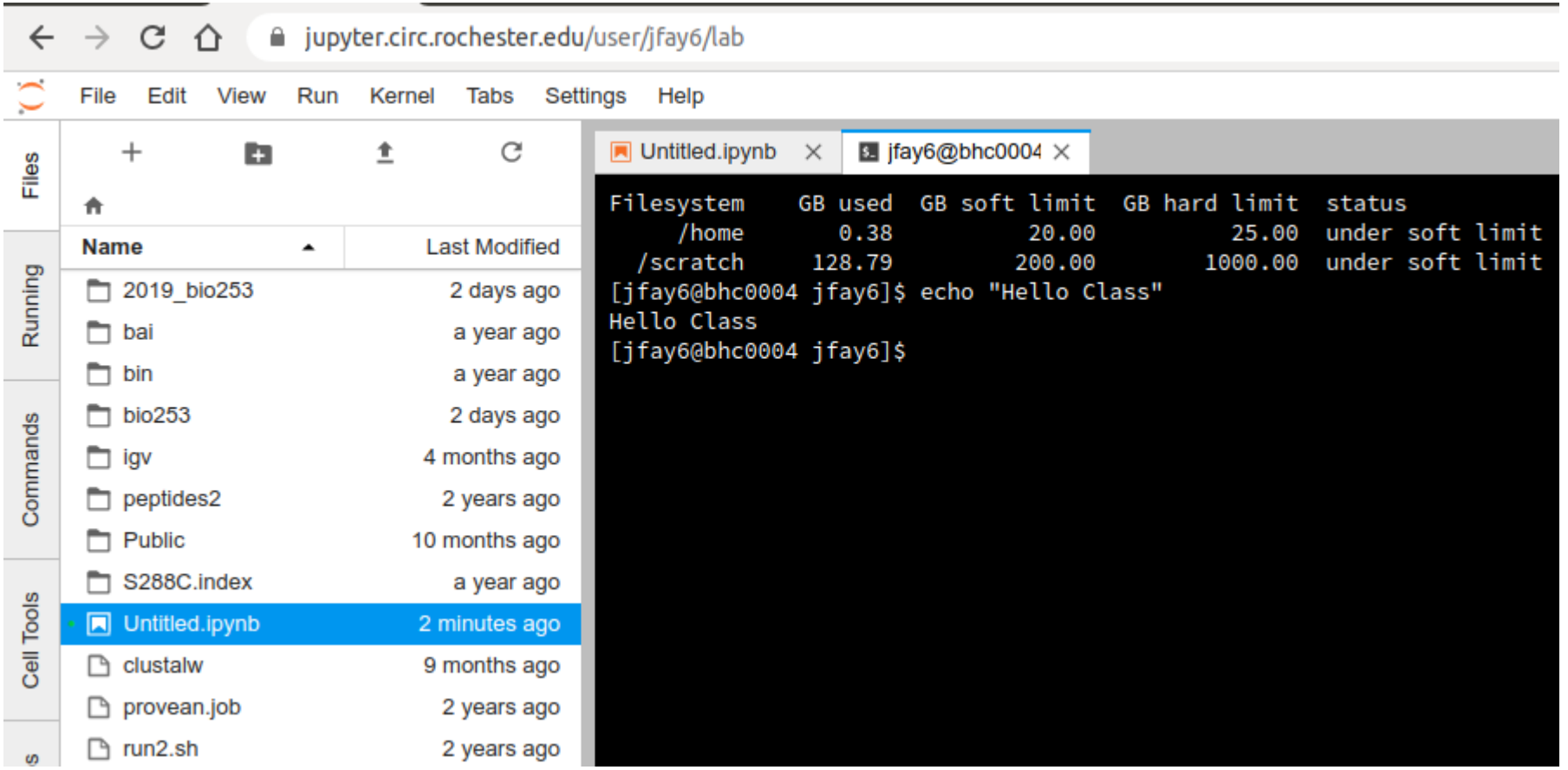
- Bash basics (move, copy, grep)

Jupyter notebook



Files and directory are hosted on bluehive
Completed labs can by exported as HTML, then save as PDF, and uploaded to Gradescope

Jupyter Terminal



Use the terminal for Lab 01. Submit answers as a PDF document uploaded to gradescope. NB the terminal may not be available using Jupyter installed on Windows

Shutdown Jupyter session rather than just closing the tab, otherwise your session won't be terminated.

Bio253W
Writing

See Blackboard for writing assignments, due dates.

Timeline

January 31 - Chose software and database
February 14 - Software draft due
February 28 - Software revision due
April 4 - Database draft due
April 18 - Database revision due

The final 395W grade

25 points: Software package draft
25 points: Software package revision
25 points: Database draft
25 points: Database revision

Two reports (database, software)

- 800 and 1200 words
- Summary
- Background and significance:
- Data
- Methods
- Intended use and example
- Conclusions and comparisons
- Literature cited

Ten simple rules for biologists learning to program

Rule 1: Begin with the end in mind
Rule 2: Baby steps are steps
Rule 3: Immersion is the best learning tool
Rule 4: Phone a friend
Rule 5: Learn how to ask questions
Rule 6: Don’t reinvent the wheel
Rule 7: Develop good habits early on
Rule 8: Practice makes perfect
Rule 9: Teach yourself
Rule 10: Just do it

Carey MA, Papin JA (2018) Ten simple rules for biologists learning to program. PLoS Comput Biol 14(1): e1005871. https://doi.org/10.1371/journal.pcbi.1005871