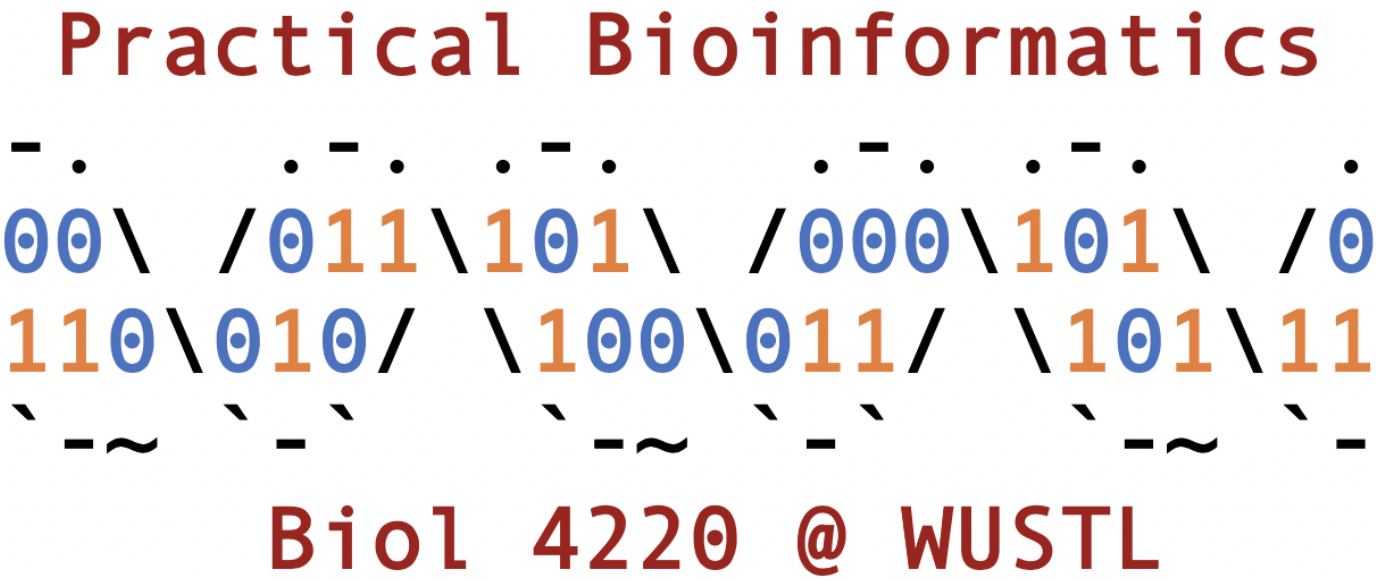
Lecture 01

Intro to practical bioinformatics



Course: Practical Bioinformatics (BIOL 4220)
Instructor: Michael Landis
Email: michael.landis@wustl.edu

Practical bioinformatics

Practice foundational computing skills
for everyday biological research

We all have different backgrounds,
research interests, goals, etc.

Practical bioinformatics

Broad goals:

- Learn new computer skills for biological data analysis
- Translate research ideas into code
- Solve problems independently
- Communicate in technical terms

Practical bioinformatics

Specific skills we'll develop:

- Write and debug programs
- Build your own analysis pipeline
- Test hypotheses with pipelines
- Make reproducible research
- Communicate research findings

Instructors

Instructor

Michael Landis

michael.landis@wustl.edu

https://landislab.org

Teaching assistant

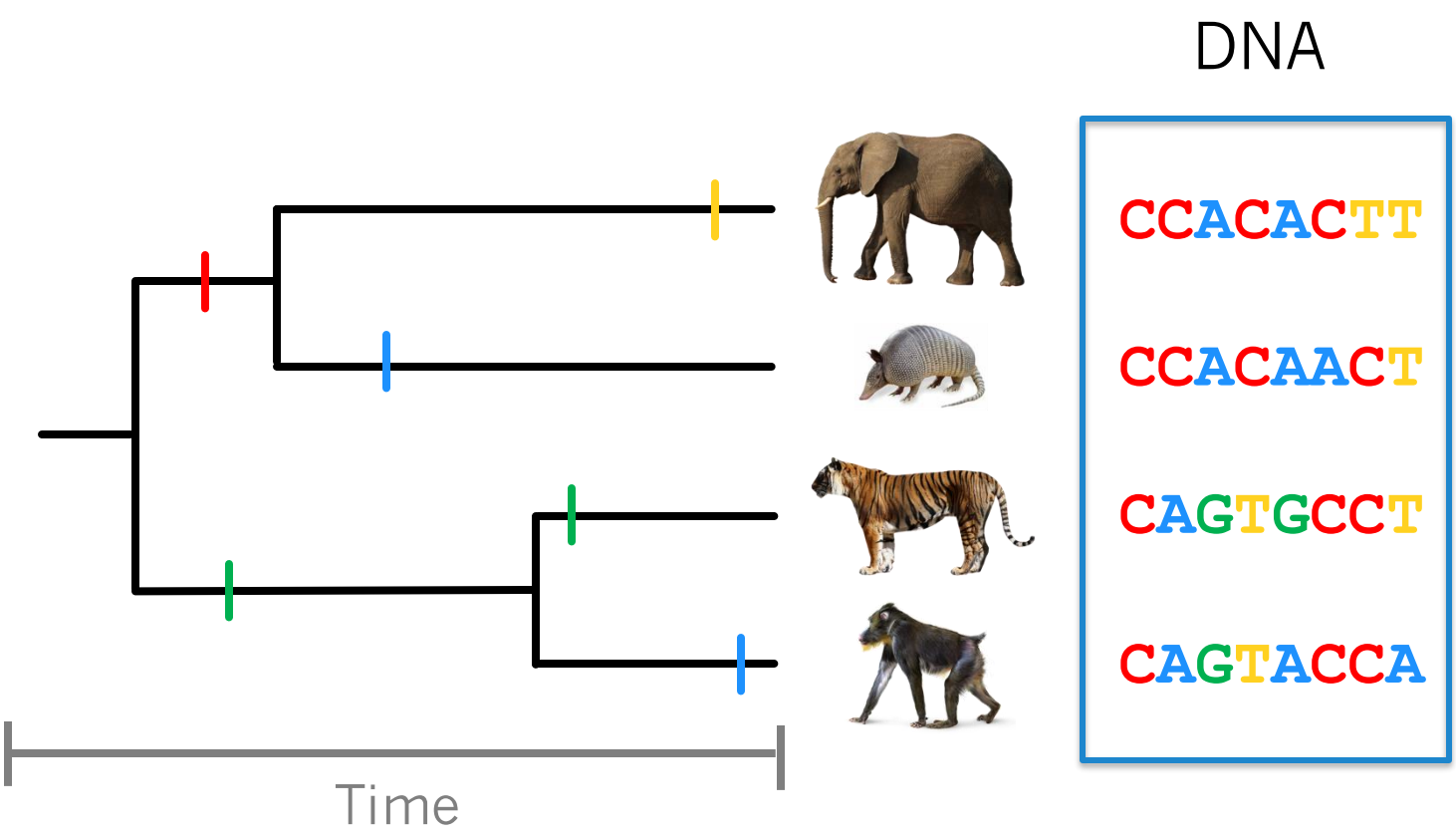
Jillian Cieslik

c.jillian@wustl.edu

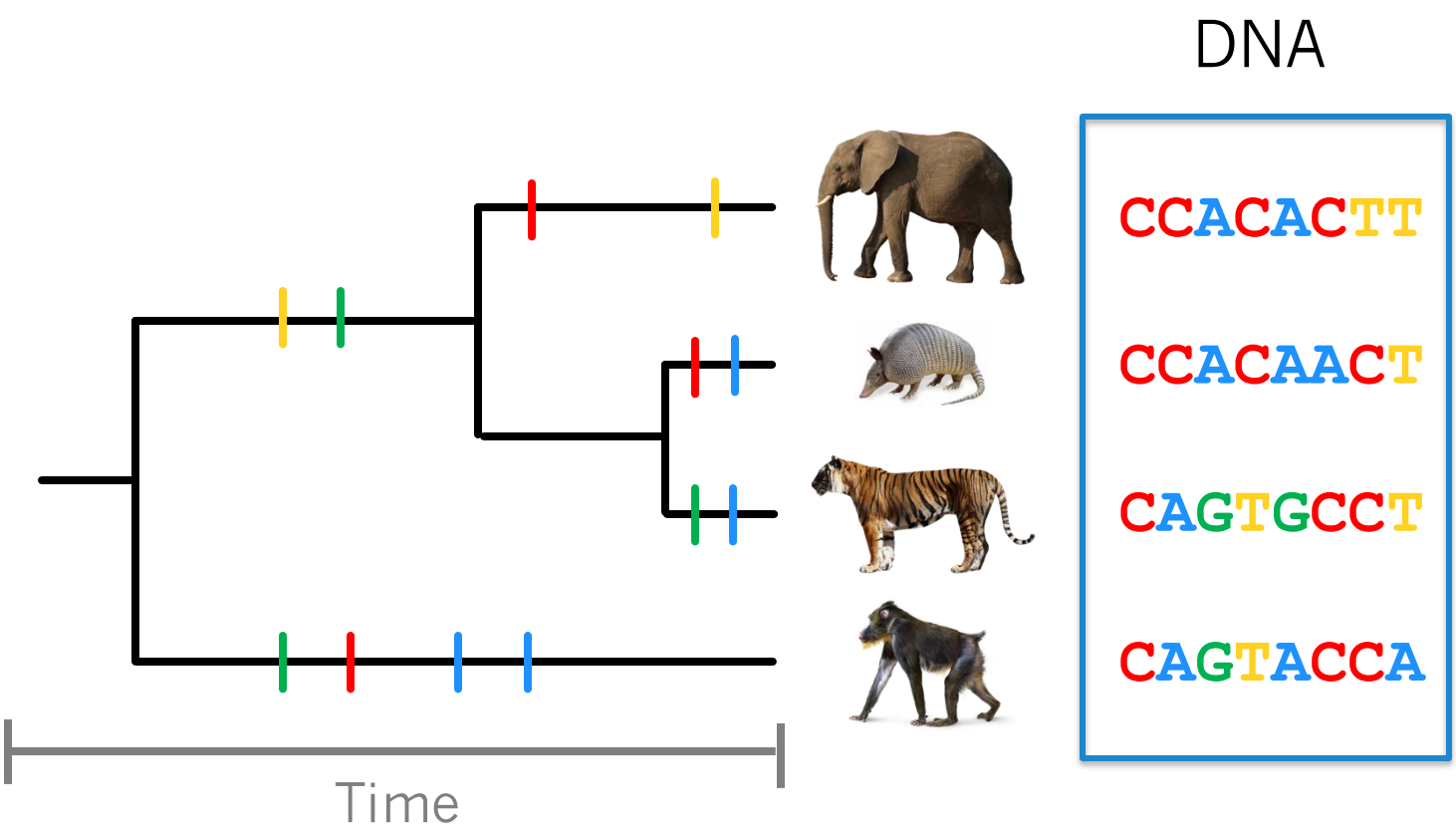
Statistical phylogenetics




How species related?
How old are they?
How did their traits evolve?



A likely history
(six mutations)



A less likely history
(twelve mutations)

Lecture 01 outline

Why bioinformatics?

Biol 4220 overview

Biol 4220 logistics

Brief intro to Unix

Why bioinformatics?

Computers are essential to modern biological research

- $1K human genome
- global biodiversity health
- human brain connectome
- tracking SARS-CoV-19
- identifying genetic diseases
- reconstructing tree of life

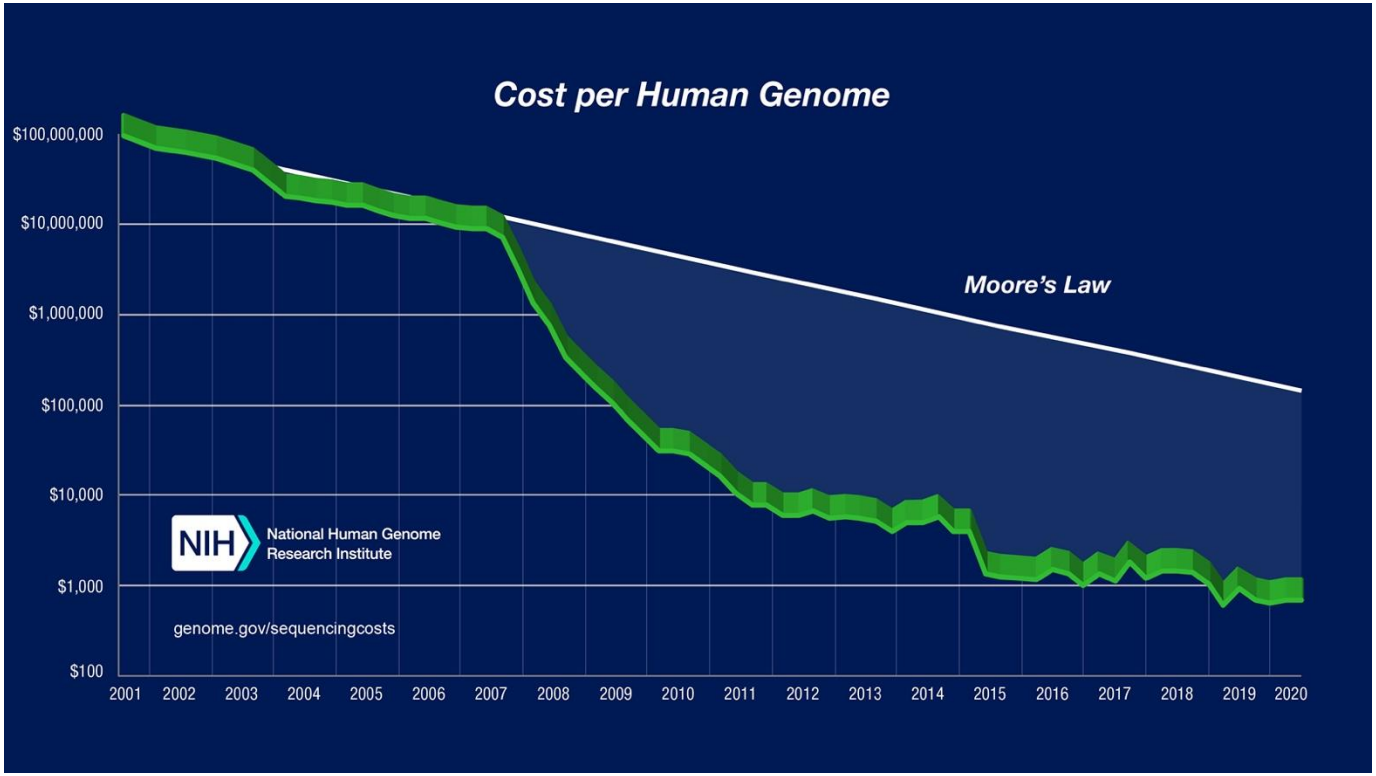
Why bioinformatics?

Different biological disciplines face similar computational challenges

Every year

- more data samples
- higher dimensional data
- methods grow more resource-intensive
- methods are more scalable
- methods are more interconnected
- need for reproducibility increases

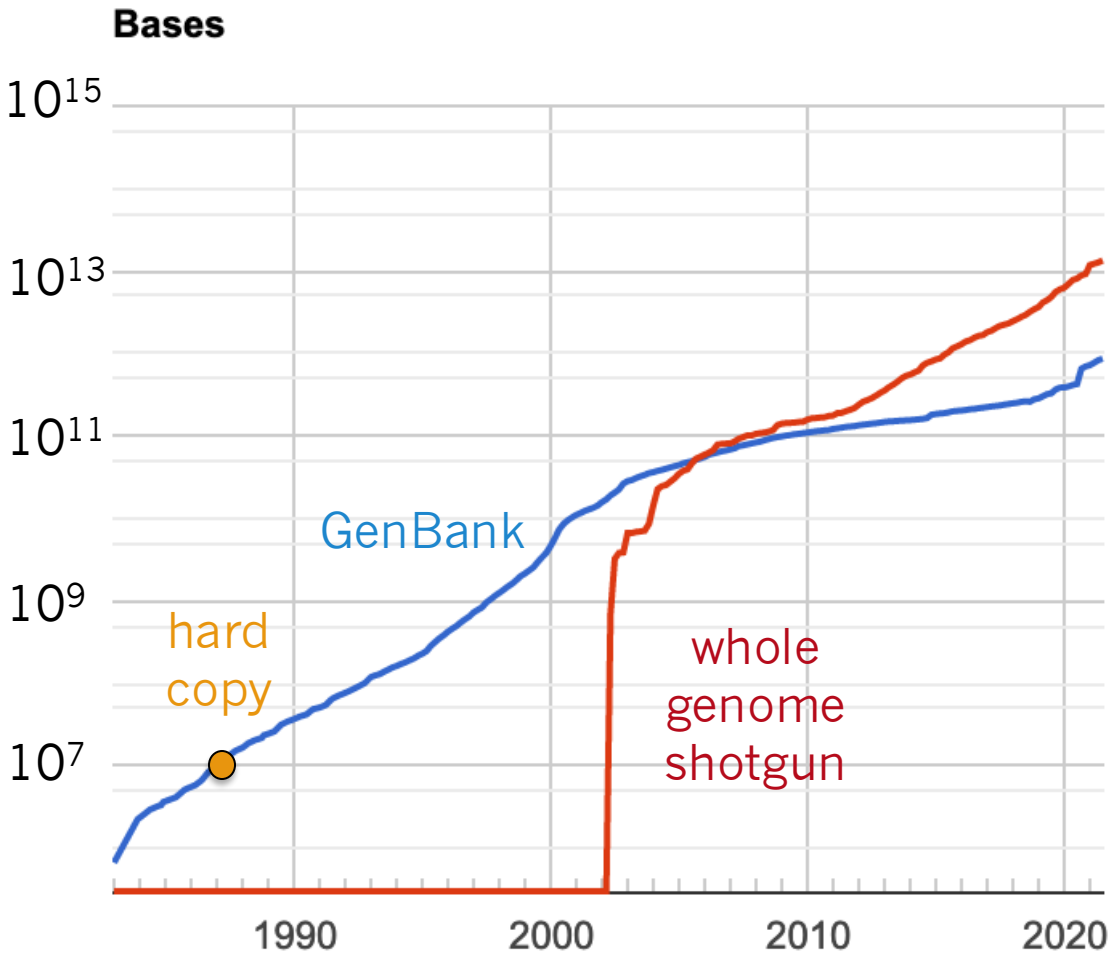
More data samples



Genome cost fall by >50% every two years

https://www.genome.gov/about-genomics/fact-sheets/Sequencing-Human-Genome-cost

More data samples



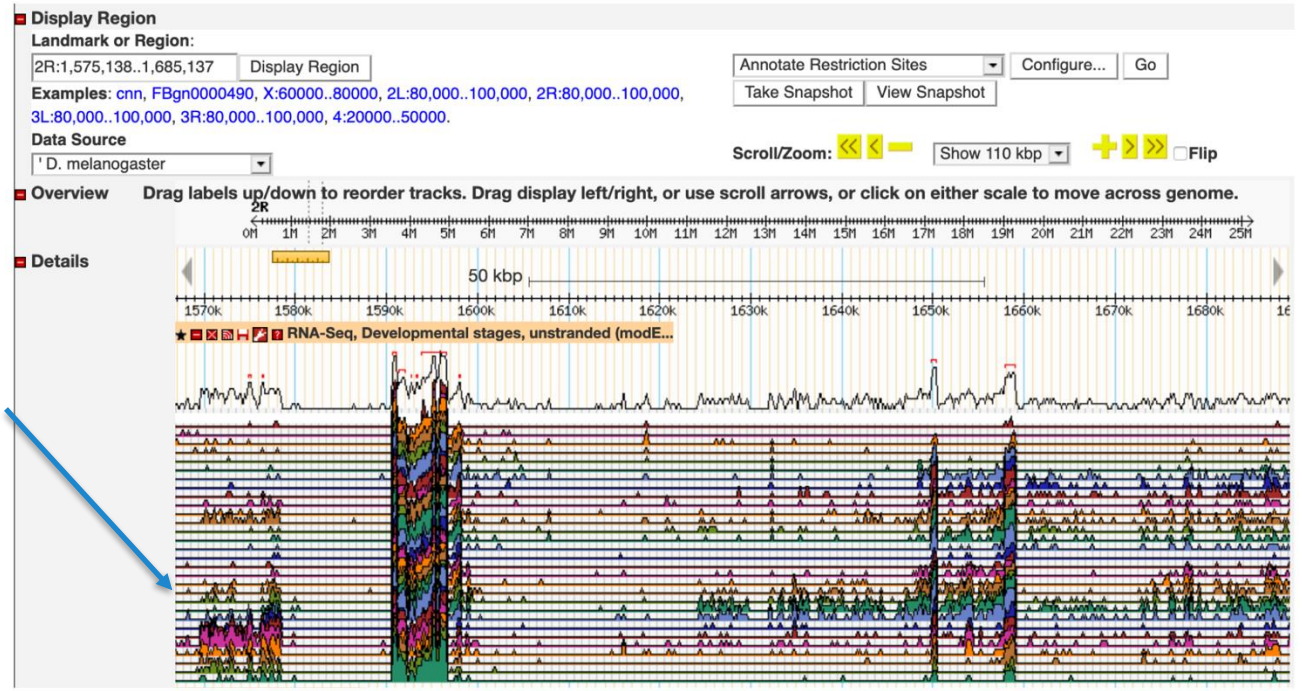
1987 hard copy of GenBank + EMBL

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

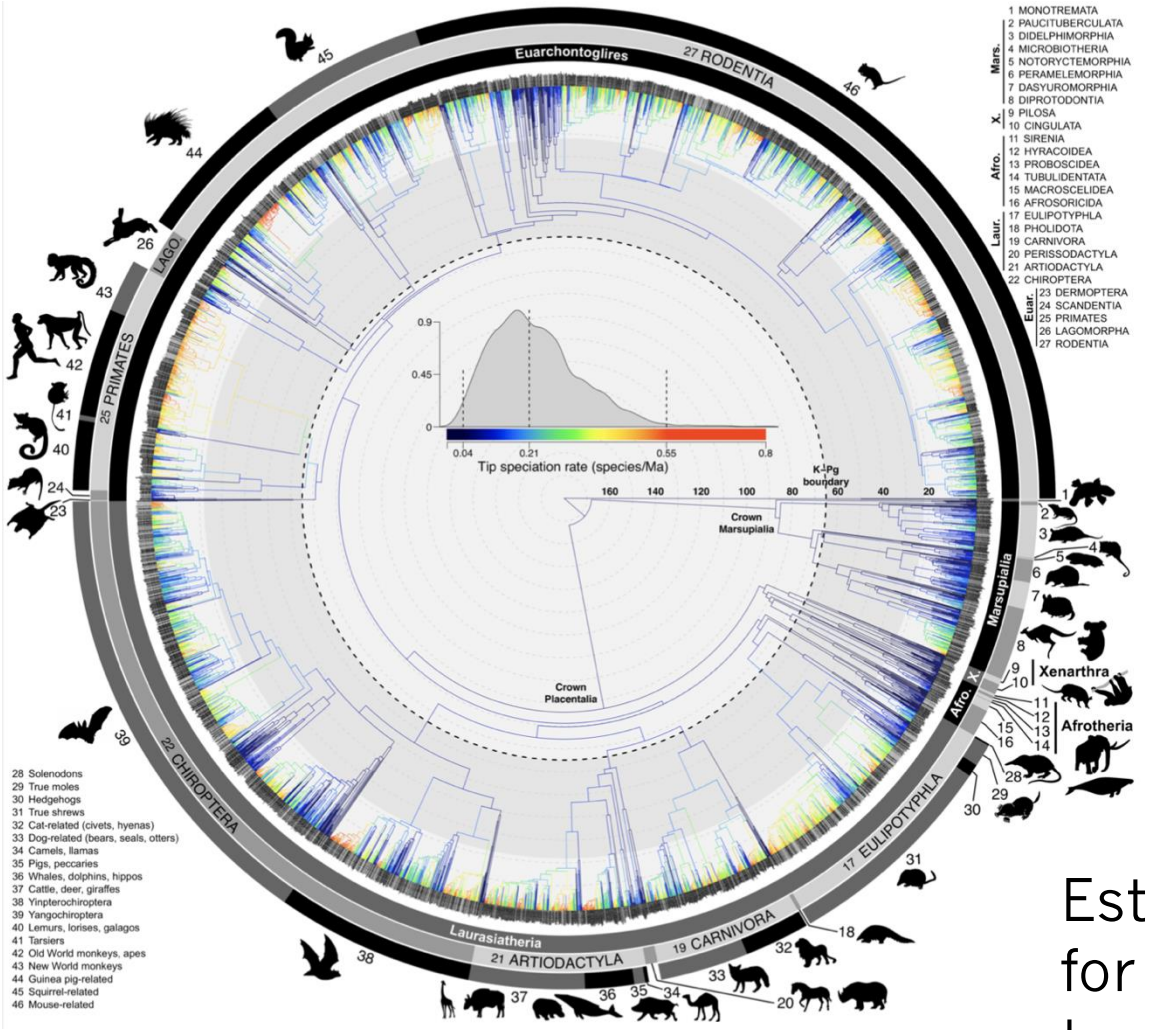
More data dimensions

Gene expression for fly

level x *gene* x *developmental stage*



FlyBase GBrowse2



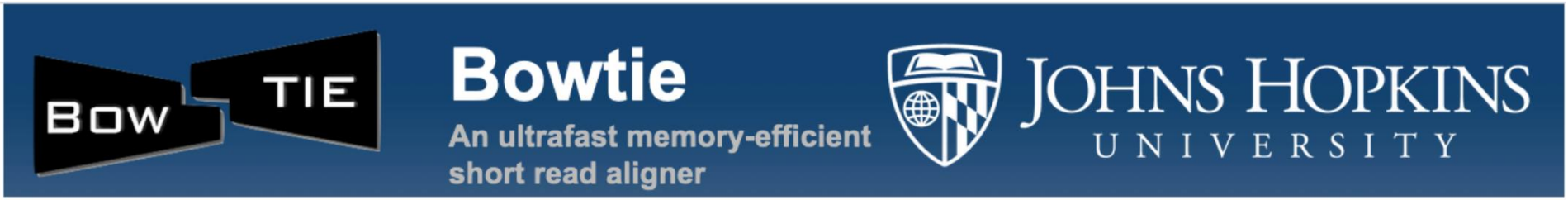
More resource-intensive methods

Estimating the phylogeny for 6000+ mammal species took 120+ computer years

Upham et al. (2019, PLoS Biology)

More scalable methods

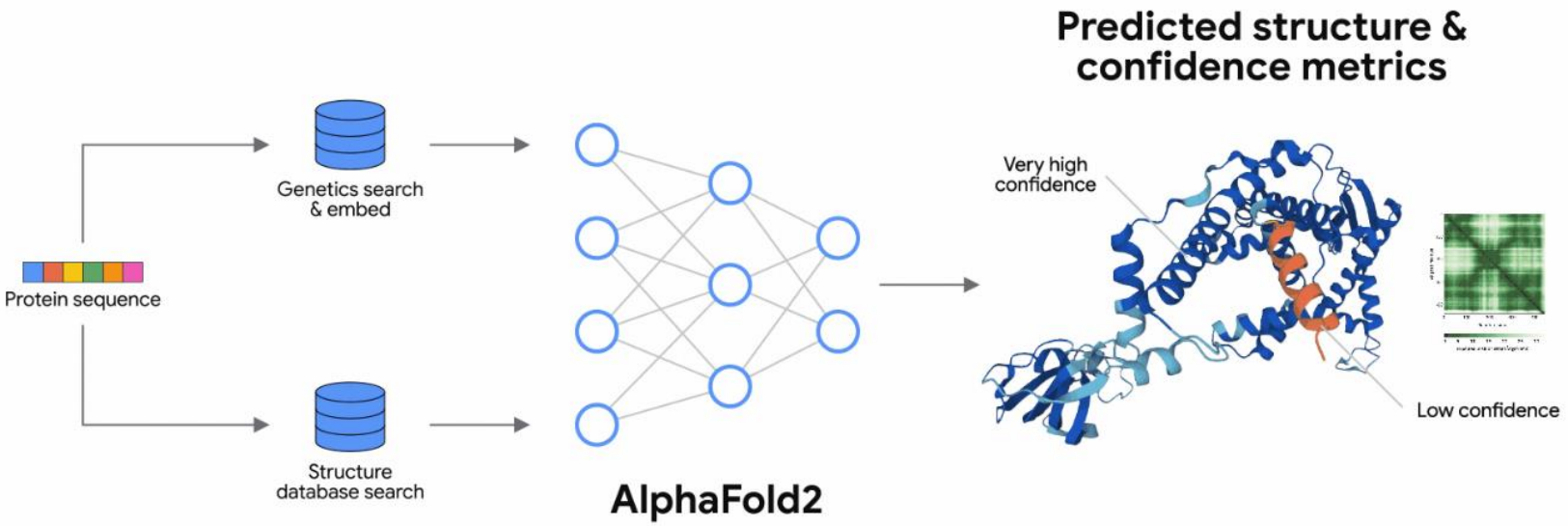
25 million 35-bp reads per hour
3.2 Gbp in human genome



Bowtie is an ultrafast, memory-efficient short read aligner. It aligns short DNA sequences (reads) to the human genome at a rate of over 25 million 35-bp reads per hour. Bowtie indexes the genome with a Burrows-Wheeler index to keep its memory footprint small: typically about 2.2 GB for the human genome (2.9 GB for paired-end).

Bowtie: short-read alignment software

More detailed predictions



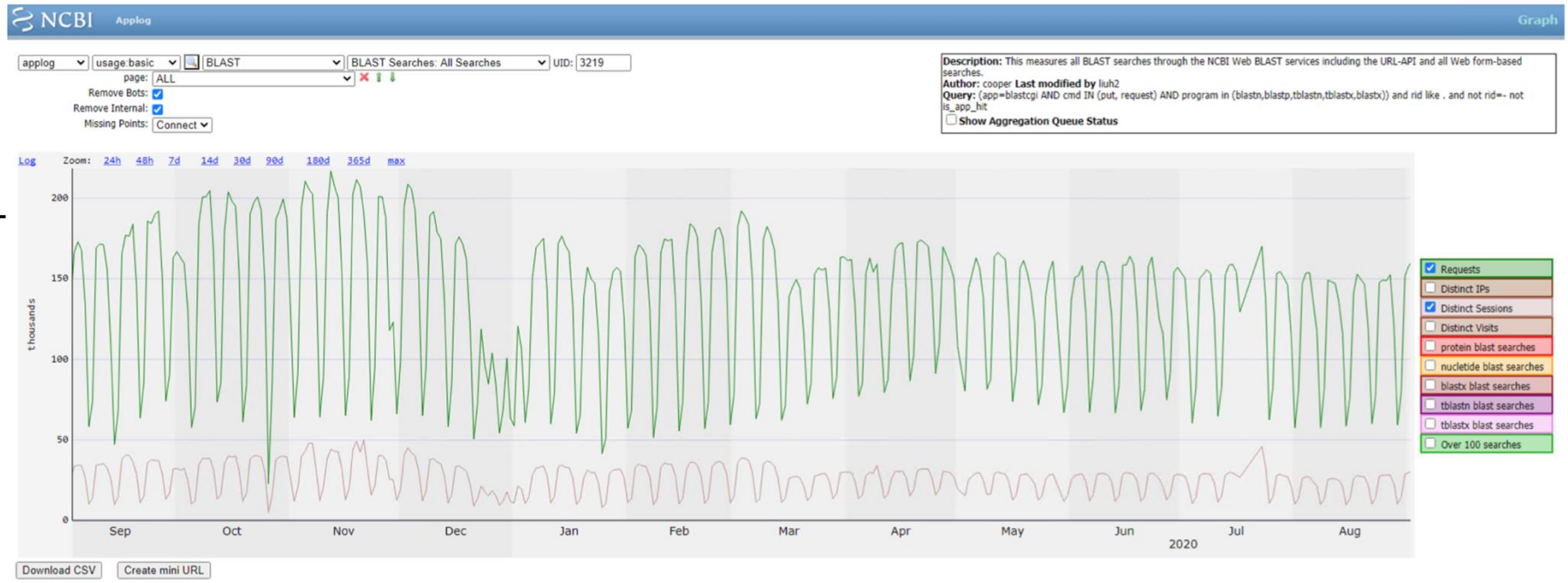
Trained with:

- ~170k known protein sequences
- ~200M known protein sequences

Google's AlphaFold2

More interconnected methods

NCBI: 150k+ BLAST requests each weekday

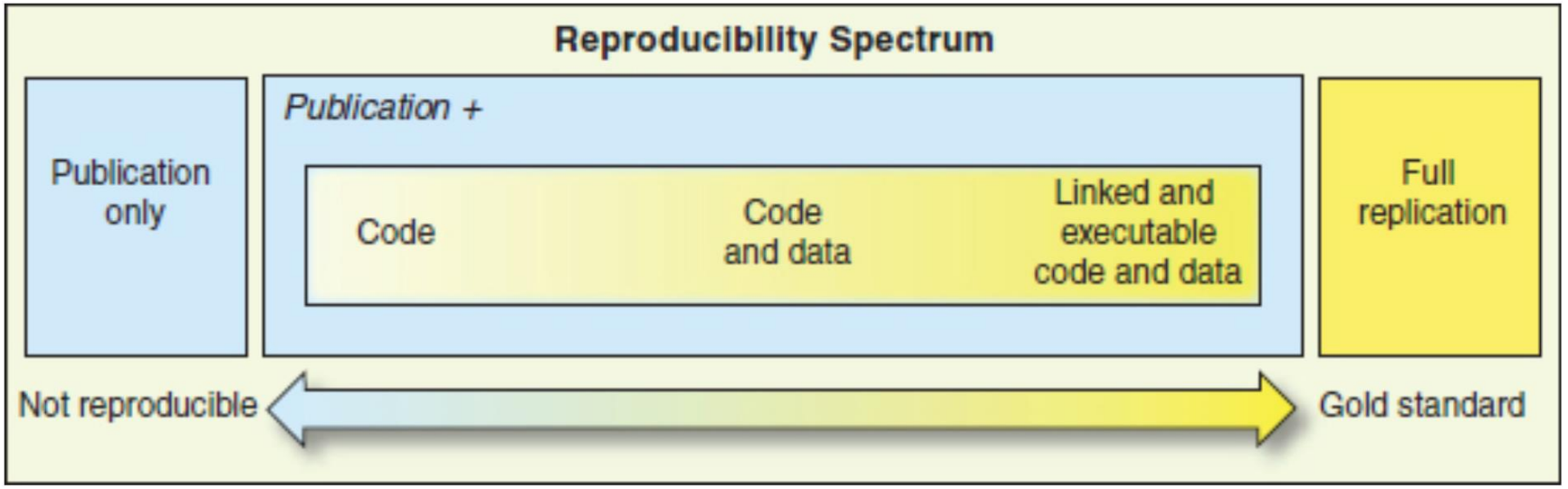


thousands of requests

provided by john.sullivan@nih.gov

Greater need for reproducibility

Computational methods allow exact reproduction of published results



from “Reproducible research in computational science” in Peng (2011, Science)

What interests *you* in biological research?

You wrote:

- genetics and genomics
- population genetics
- phylogenetics
- gene expression
- neuroscience
- cancer biology
- pharmacology
- medicine
- immunology

Why are *you* interested in bioinformatics?

You wrote:

- curious about field
- learn new tools and techniques
- difficult research project
- improve coding skills
- develop confidence
- learn about operating systems
- preparation for future career
- build an analysis pipeline

Book recommendations

War Doctor: Surgery on the Front Line by David Nott
Project Hail Mary by Andy Weir
The Midnight Library by Matt Haig
Into the Wild by John Krakauer
Labyrinths by Jose Luis Borges
Atomic Habits by James Clear
Pachinko by Min Jin Lee
East of Eden by John Steinbeck
The House of the Spirits by Isabel Allende
A Simple Act of Violence by Roger Jon Ellory

Biol 4220 topics

Computational skills

- Unix-based operating systems
- Python and shell scripts
- scientific computing libraries
- version control software
- bioinformatics pipeline design

Biological problems

- sequence processing
- molecular phylogenetics
- hypothesis testing

Course page

All course info is centralized here:

github.com/WUSTL-BioI4220/home

Contains links to:

- syllabus
- lectures
- labs
- course project
- GitHub Classroom

Labs

Each lab focuses on a new set of skills

One lab is assigned per class

Labs will be submitted using an online tool called GitHub Classrooms

Free Writing

Reinforce understanding

Submissions earning high grades have these qualities:

- demonstrates understanding of recent material
- explains their strategy for solving recent lab assignment
- describes why they found recent material interesting or confusing
- suggests good quiz questions based on recent material

Each Monday after lecture for 15 min

Quizzes

Assess understanding

Closed notes, pen and paper

Example

Q: Describe what each step does in this command:

```
ls *.txt | grep -v "draft" | wc -l
```

A: This pipeline (1) lists all local files ending in “txt”,
(2) filters out files that contain “draft” in the name (*-v*), and
then (3) prints the number (*-l* for lines) of final matches

Each Wednesday after lecture for 15 min

Course project

Design a pipeline to analyze genes

Example pipeline:

- download and align sequences
- summarize molecular variation
- build molecular phylogeny
- print and plot output
- add 2+ unique features

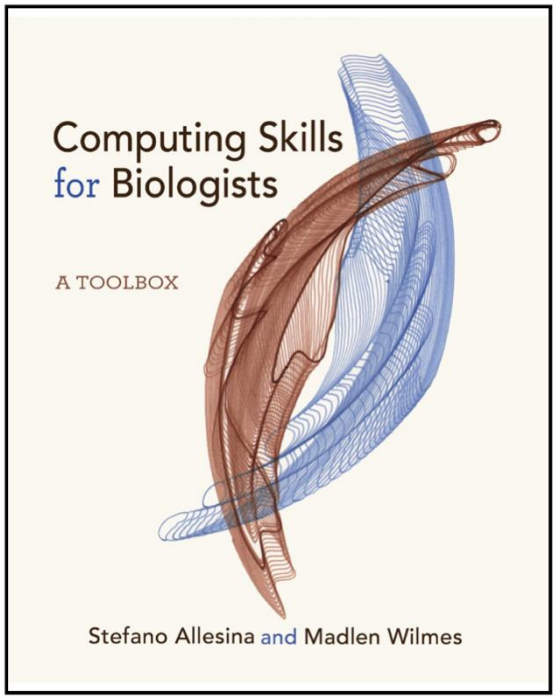
At the end of the semester, students will

- present their pipeline
- submit code, output, documentation

(more details later in semester)

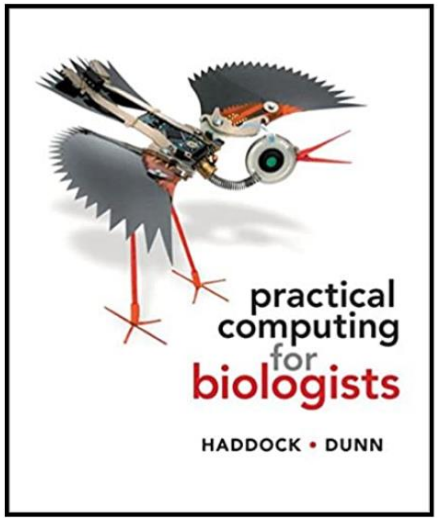
Coding resources

Primary text

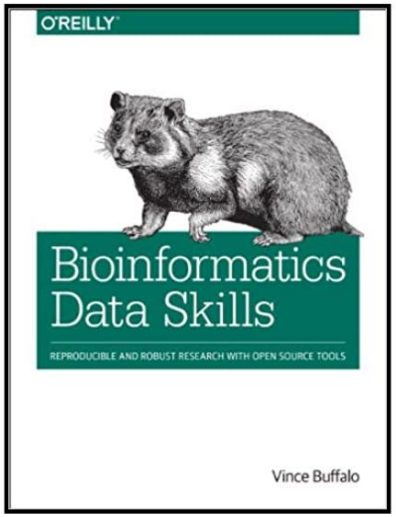


Allesina & Wilmes
ISBN: 9780691167299
~$35

...other useful texts



Haddock & Dunn
ISBN: 0878933913
~$60



Buffalo
ISBN: 1449367372
~$35

Coding resources

You may use ChatGPT, StackOverflow, etc. for Lab and Project assignments

Important to use these tools responsibly, especially when learning the fundamentals

Comment code/text using these tools

- provide example of query or search terms
- describe any errors they produced
- describe motivation to use the tool (e.g. what was confusing?)

Visit https://it.wustl.edu/ai/ for free ChatGPT access

Participation

Communicate with others!

Examples

- asking and/or answering questions
- working in groups
- helping other students
- visiting office hours
- discussing research problems

Questions?

Operating systems (OS)

Operating systems coordinate user commands with computational resources & hardware

Examples:

- Windows
- Mac OS X (Unix-based)
- Linux (Unix-based)

Most scientific computing uses ***Unix***-based systems; we'll be using the Linux distribution, ***Ubuntu***

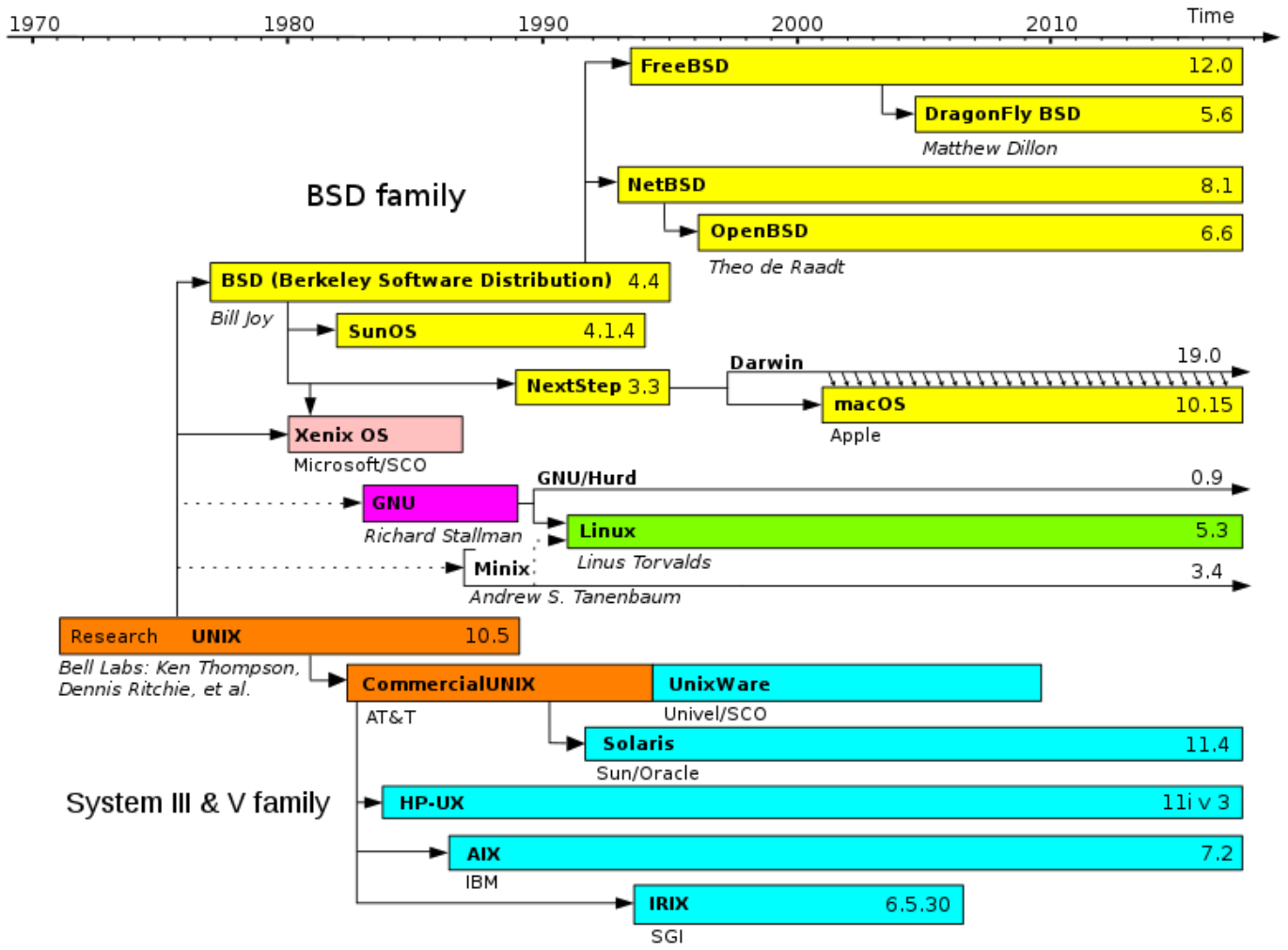
Operating systems (OS)

What do operating systems do?

They manage

- ***user interface*** for computer input/output
- ***scheduled tasks*** across multiple users/resources
- ***user interruptions*** of scheduled tasks
- ***memory use*** in efficient manner
- ***filesystem organization*** on hard drive
- ***user permissions*** for resource security
- ***network communication*** with other devices
- ***custom software*** to interact w/ OS and hardware

Unix family tree

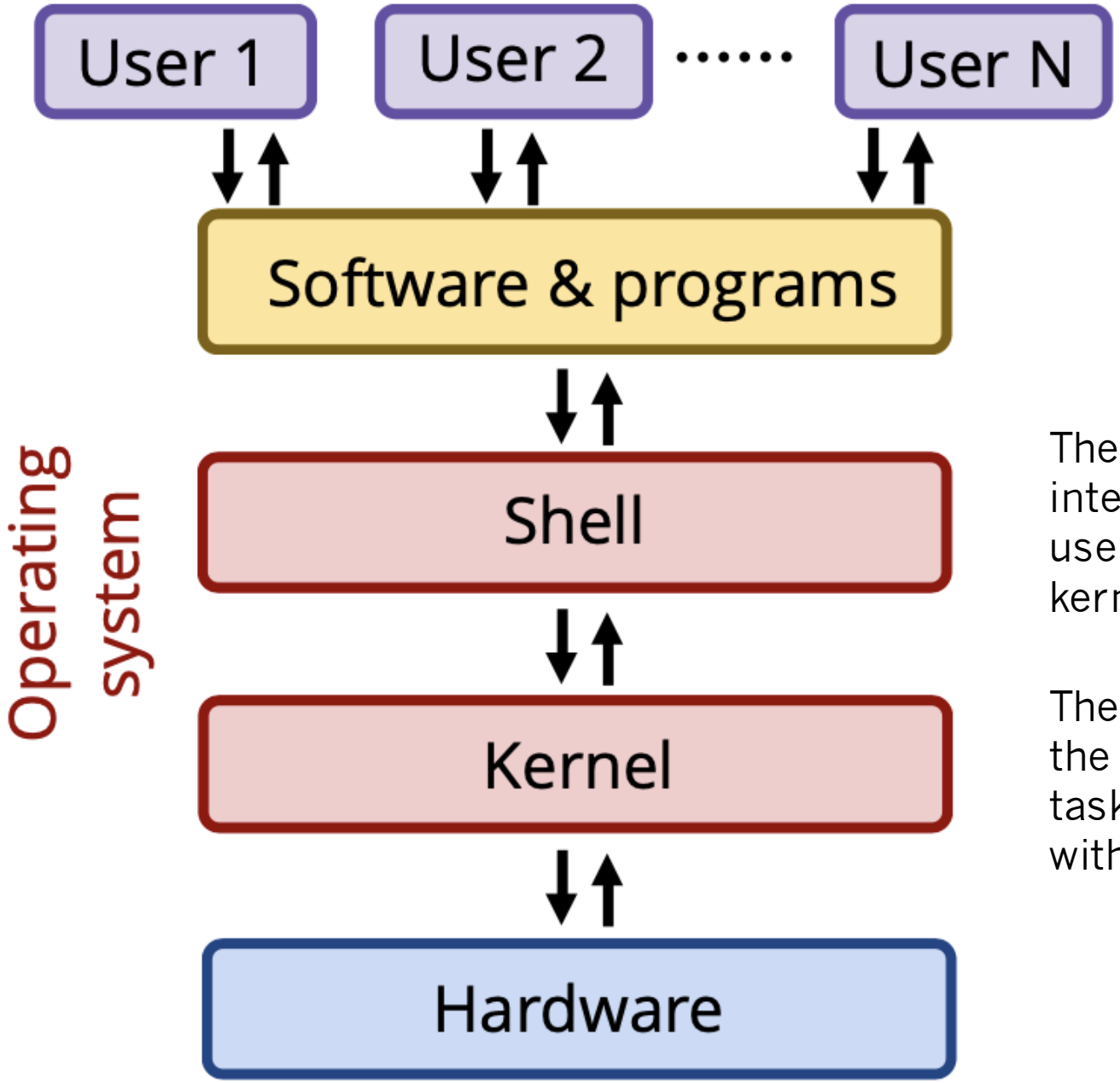


https://commons.wikimedia.org/wiki/File:Unix_timeline.en.svg

Ubuntu

We'll use Ubuntu 24.04 LTS

- Reliable testing + release cycles
- Excellent tutorials
  https://ubuntu.com/tutorials
- Extremely active support community
  https://ubuntuforums.org
- modified ***Debian kernel*** for stability
- popular ***bash shell*** by default



The ***shell*** is the interactive interface between the users, programs, and the kernel

The ***kernel*** is the core of the OS that controls all tasks and interfaces with the hardware

Kernel vs. shell

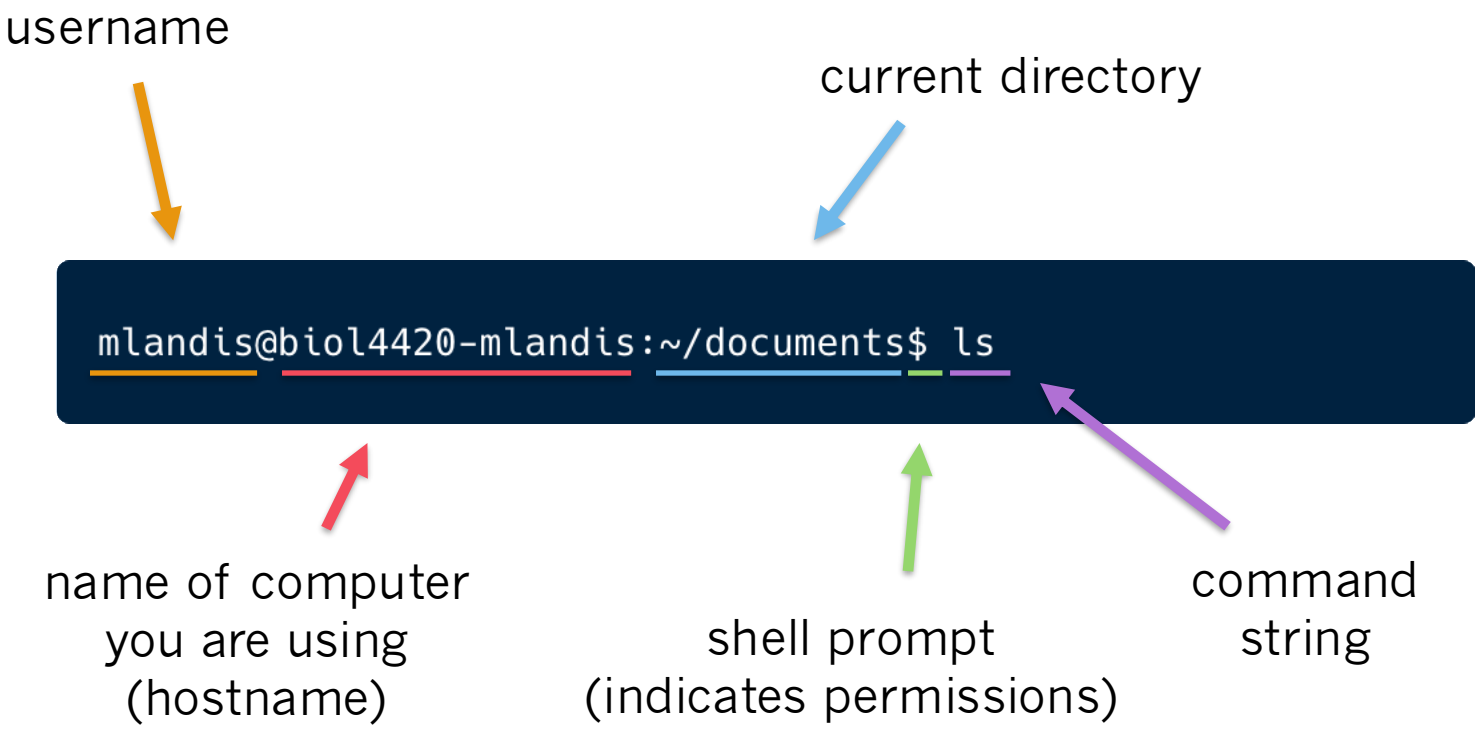
The ***kernel*** has control over all computer resources, including processors, memory, storage, devices, task management, etc.

The ***shell*** is a command line interface and scripting language that communicates user commands to the kernel for processing

```
> # connect to my workstation
> ssh mlandis@128.252.89.47
```

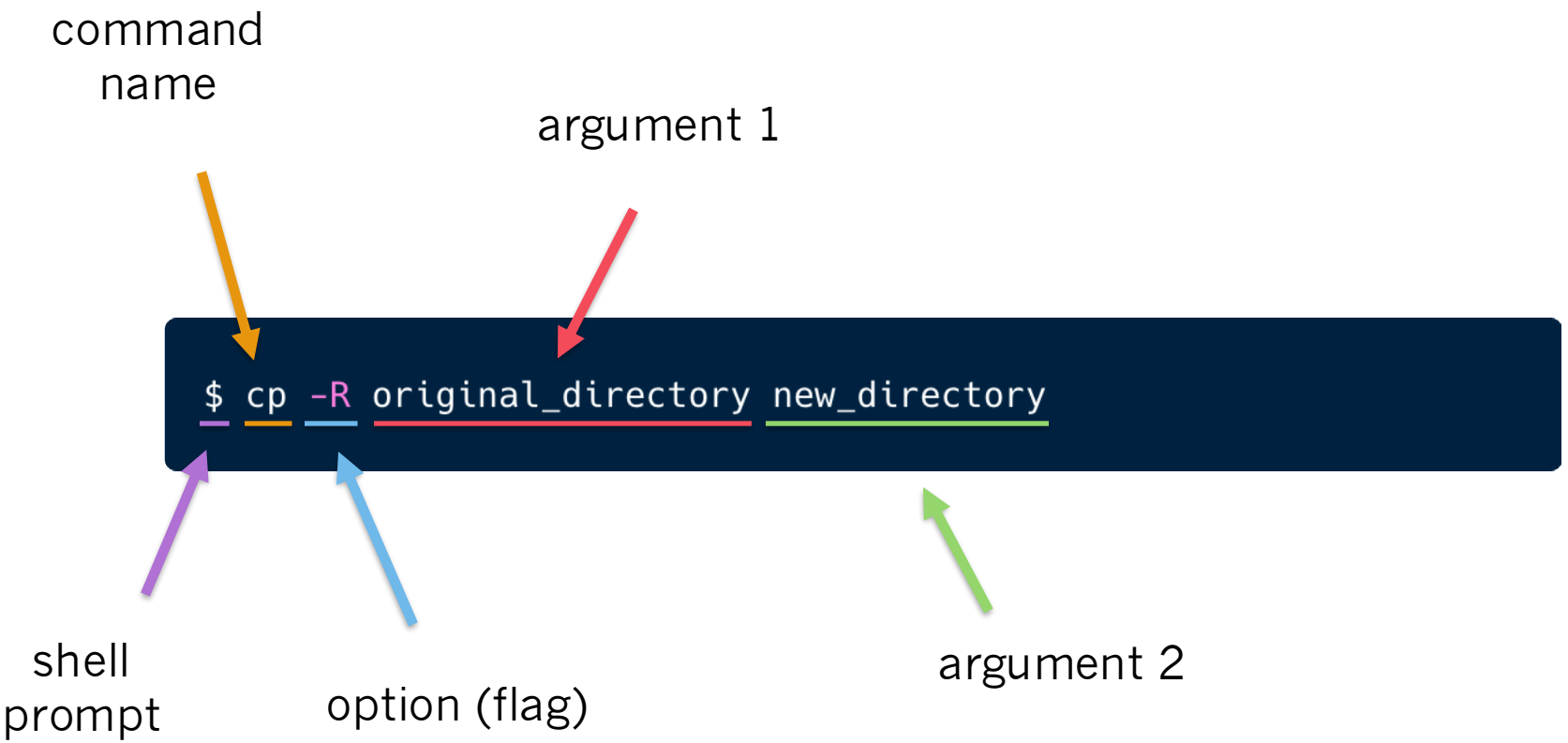
Example of Unix shell command

Command line



The ***command line*** accepts commands provided by the user (you!)

Command string

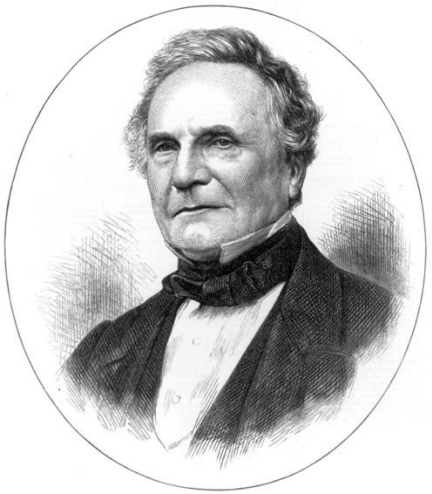


A ***command*** is applied against an ***argument(s)*** and its behavior can be modified by ***option(s)***

Computers are predictable

- accept input as data
- process that data
- output processed data

Charles Babbage
"father of the computer"



"On two occasions, I have been asked [by members of Parliament], 'Pray, Mr. Babbage, **if you put into the machine *wrong* figures, will the right *answers* come out?**' I am not able to rightly apprehend the kind of confusion of ideas that could provoke such a question."

Overview for Lab 01