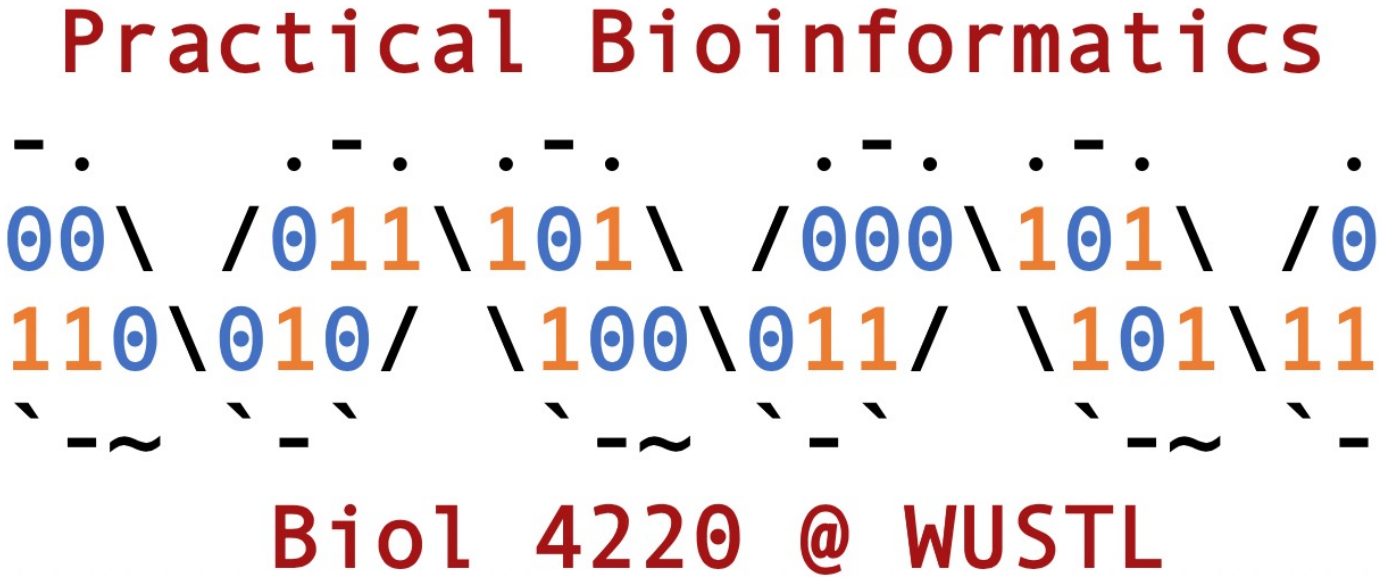
Lecture 01

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:

- Develop confidence using computers
- Translate research ideas into code
- Solve problems independently
- Communicate in technical terms
- Stay healthy

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

Class info

Lecture + Lab

Mon + Wed, 8:30am – 12:00pm

Life Science 117 (computer lab)

Office hours

Thu, 2:00pm – 4:00pm

Rebstock 210 (by appointment)

Instructors

Instructor

Michael Landis

michael.landis@wustl.edu

https://landislab.org

Graduate student instructor

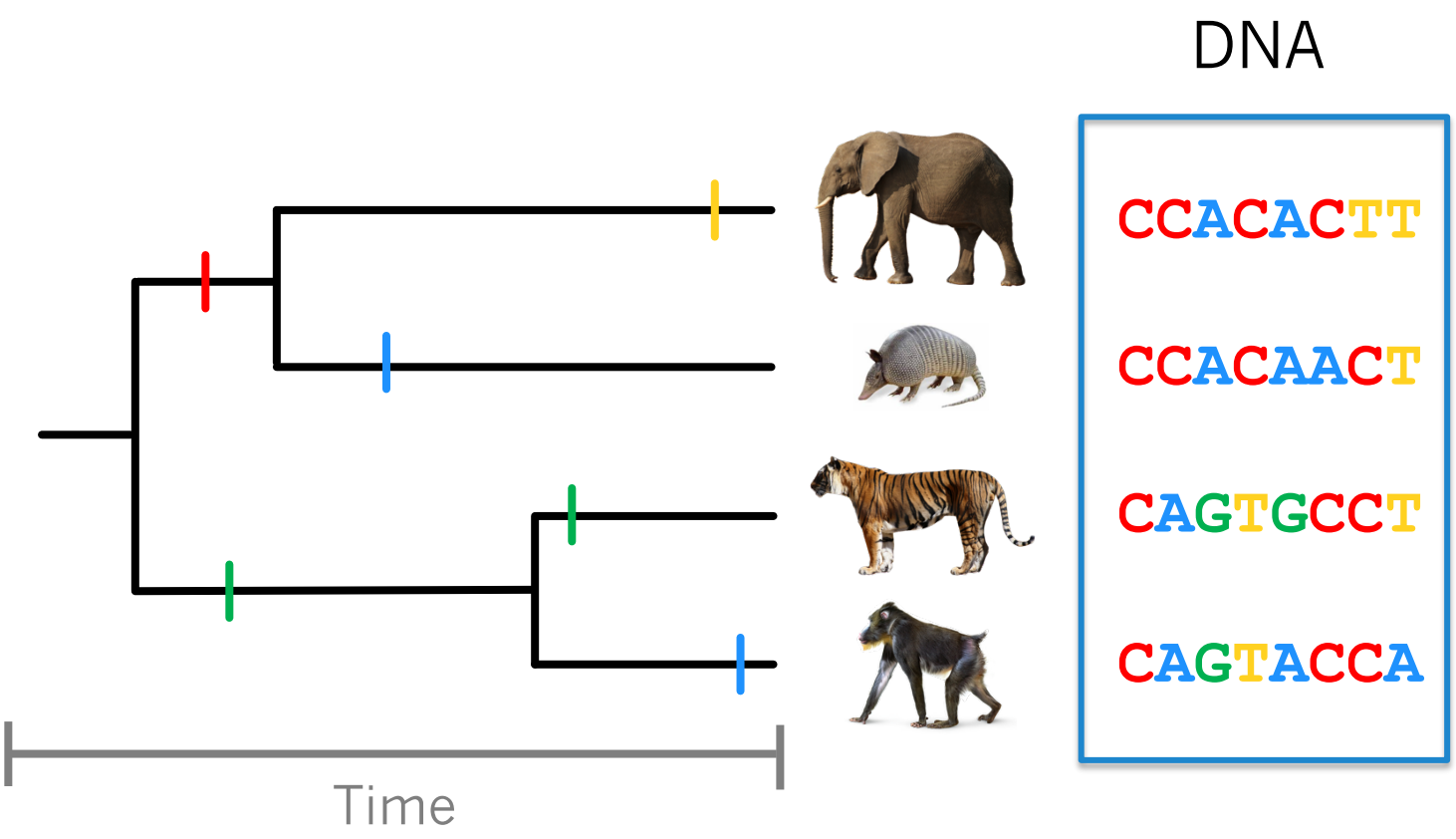
Nathan Wamsley

n.t.wamsley@wustl.edu

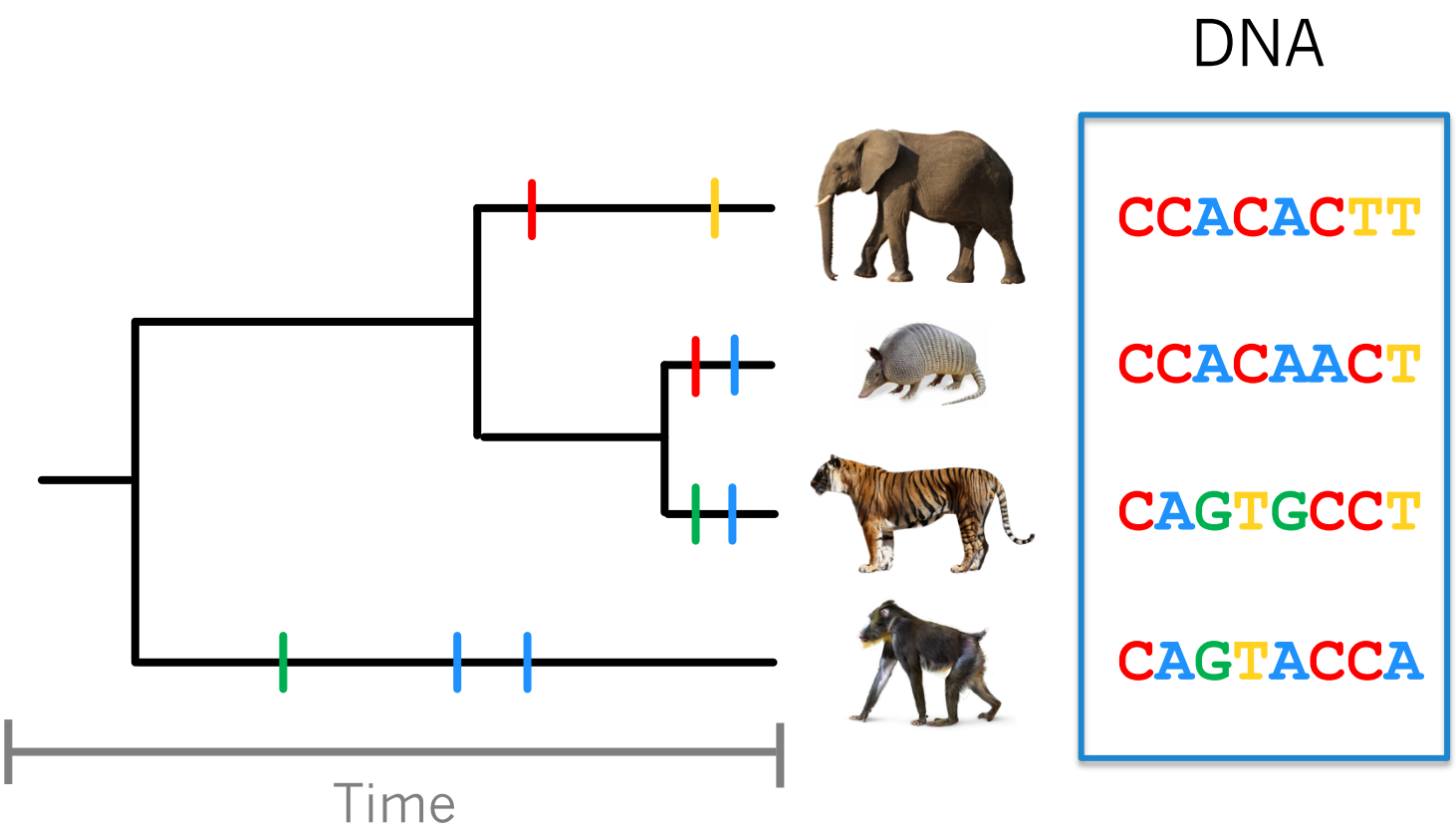
Statistical phylogenetics




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



A likely history
(six mutations)



A less likely history
(nine mutations)

Lecture 01 outline

Why bioinformatics?

Biol 4220 overview

Biol 4220 logistics

Introduction to Unix

Why bioinformatics?

Computers are essential to modern biological research

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

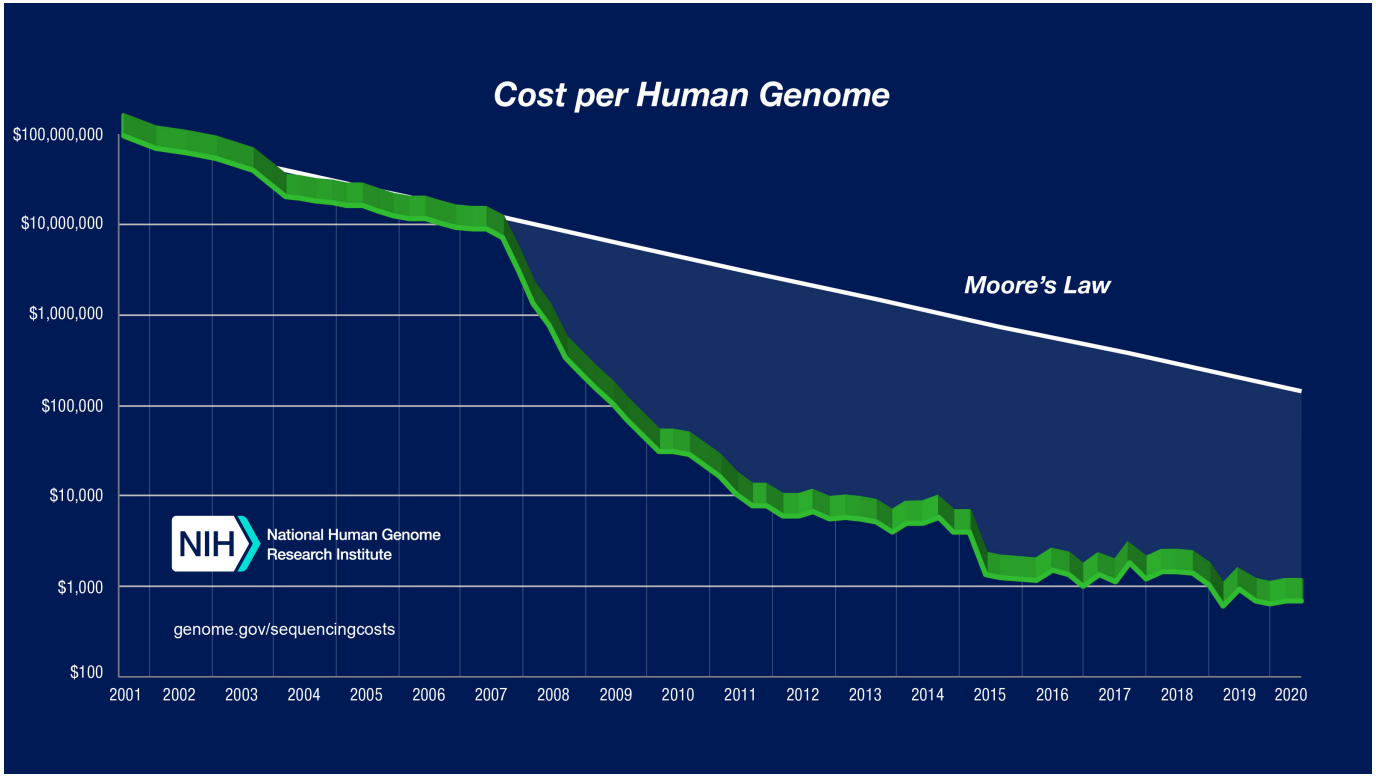
Why bioinformatics?

Different biological disciplines face similar computational challenges

Every year

- more samples in data
- higher dimensions in data
- methods are more resource-intensive
- methods are more scalable
- methods are more interconnected
- need for reproducibility increases

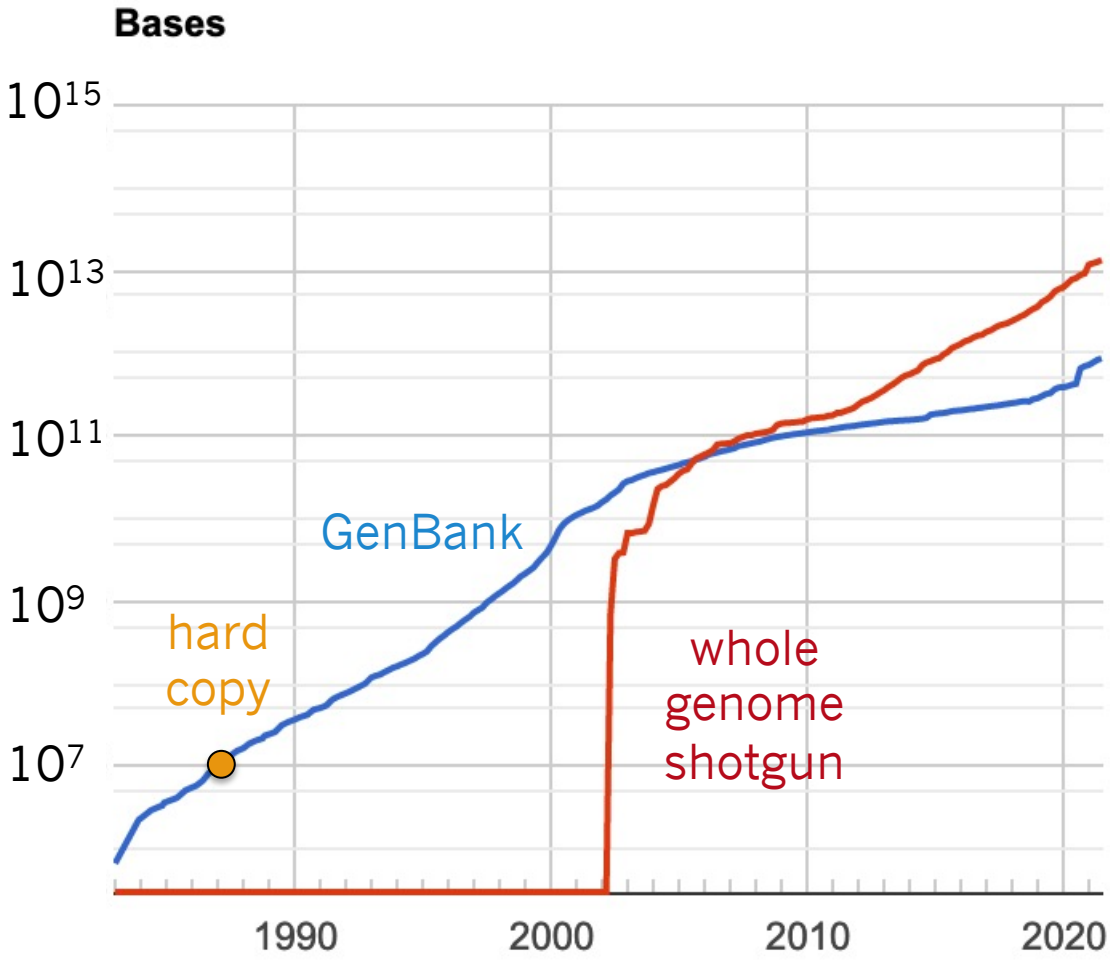
More data samples

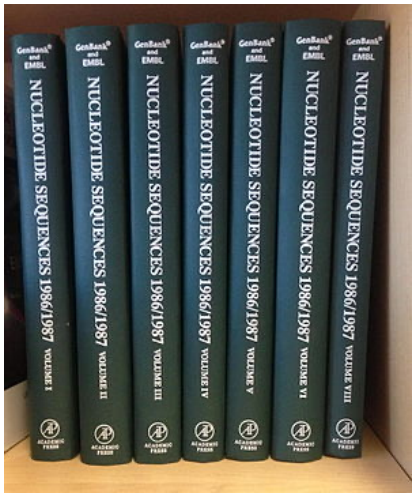


Genome cost fall by >50% every two years

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

More data samples





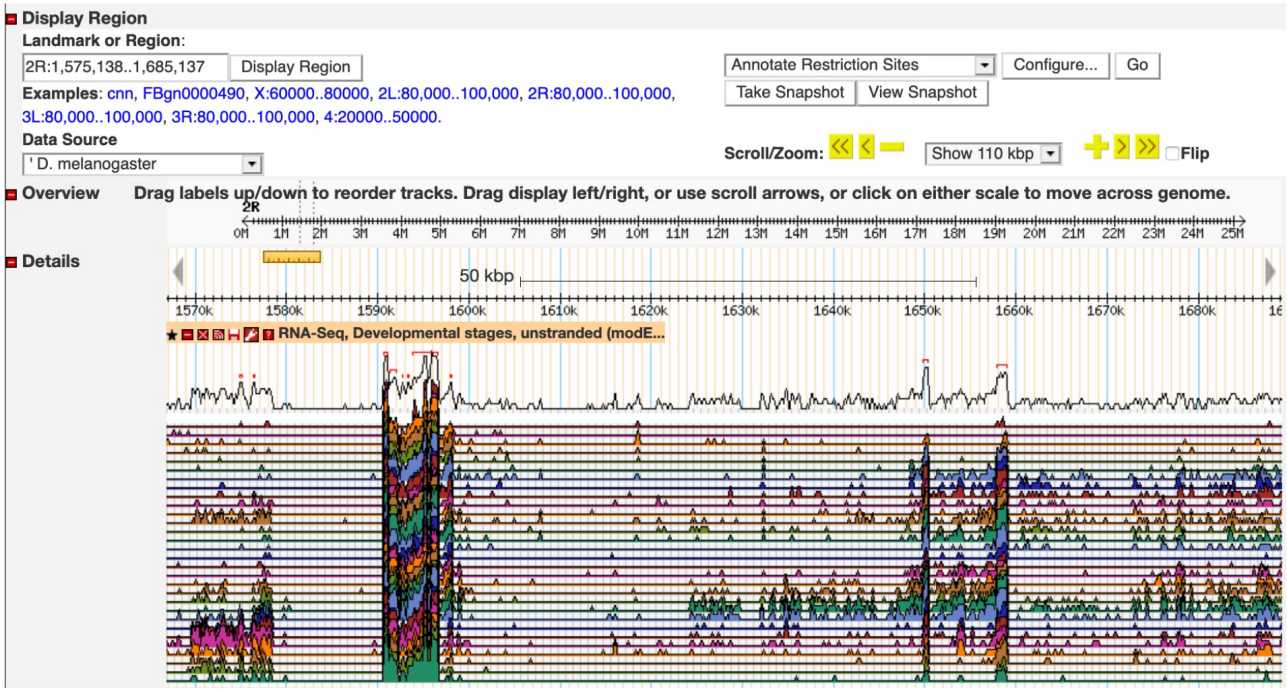
1987 hard copy of GenBank + EMBL

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

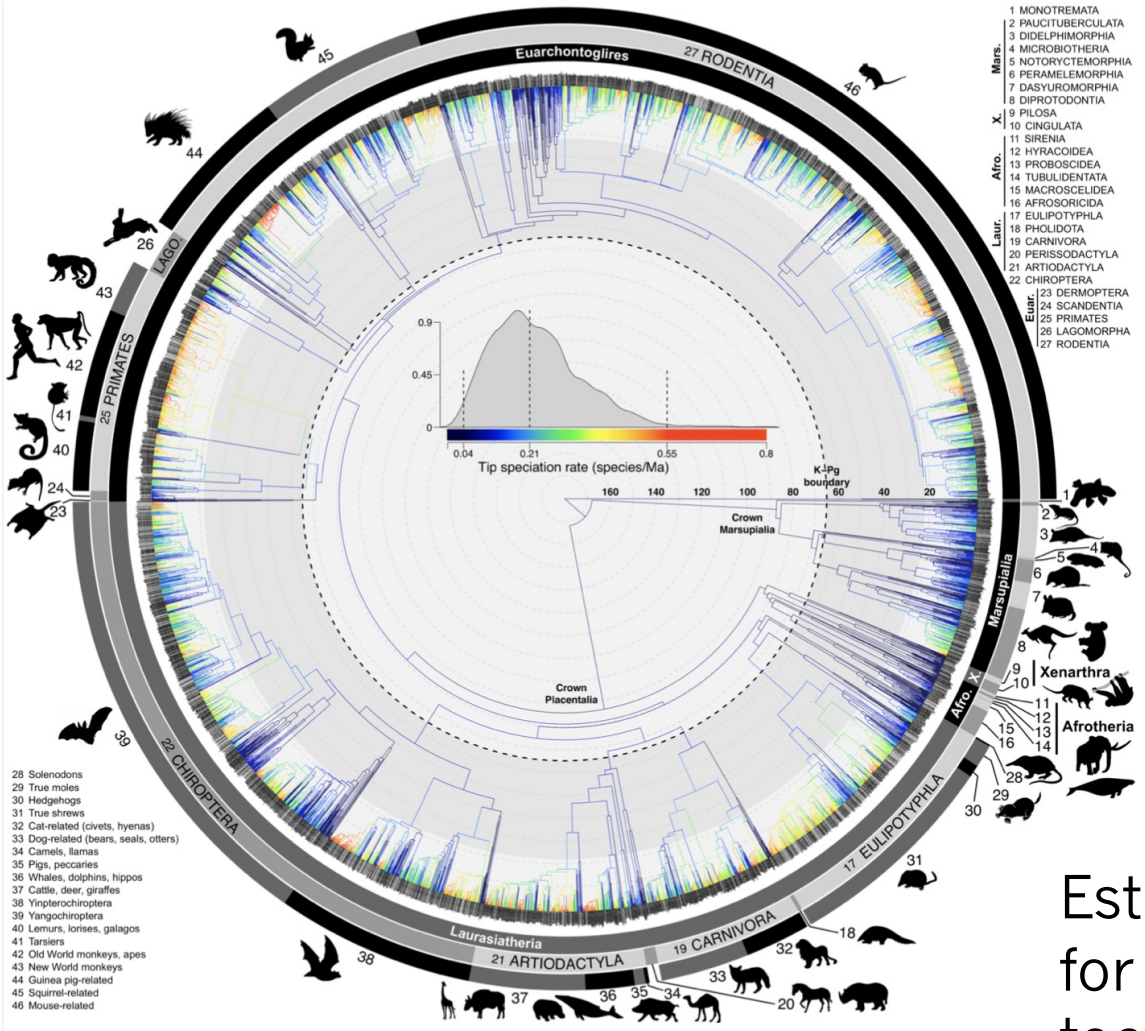
More data dimensions

Gene expression for fly

level x *gene* x *developmental stage*



FlyBase GBrowse2



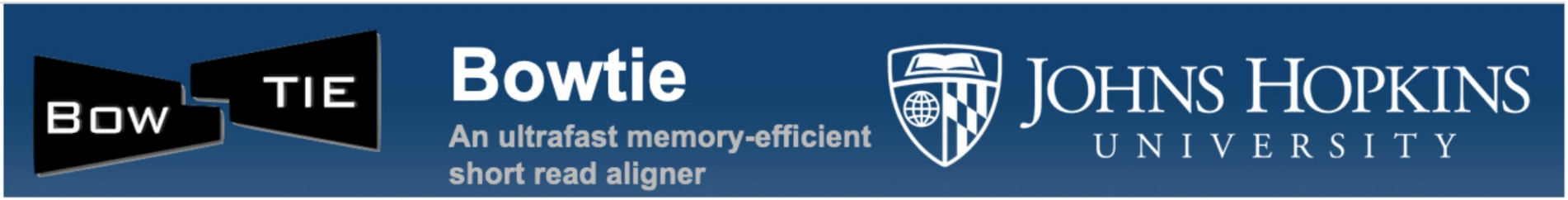
More resource-intensive methods

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



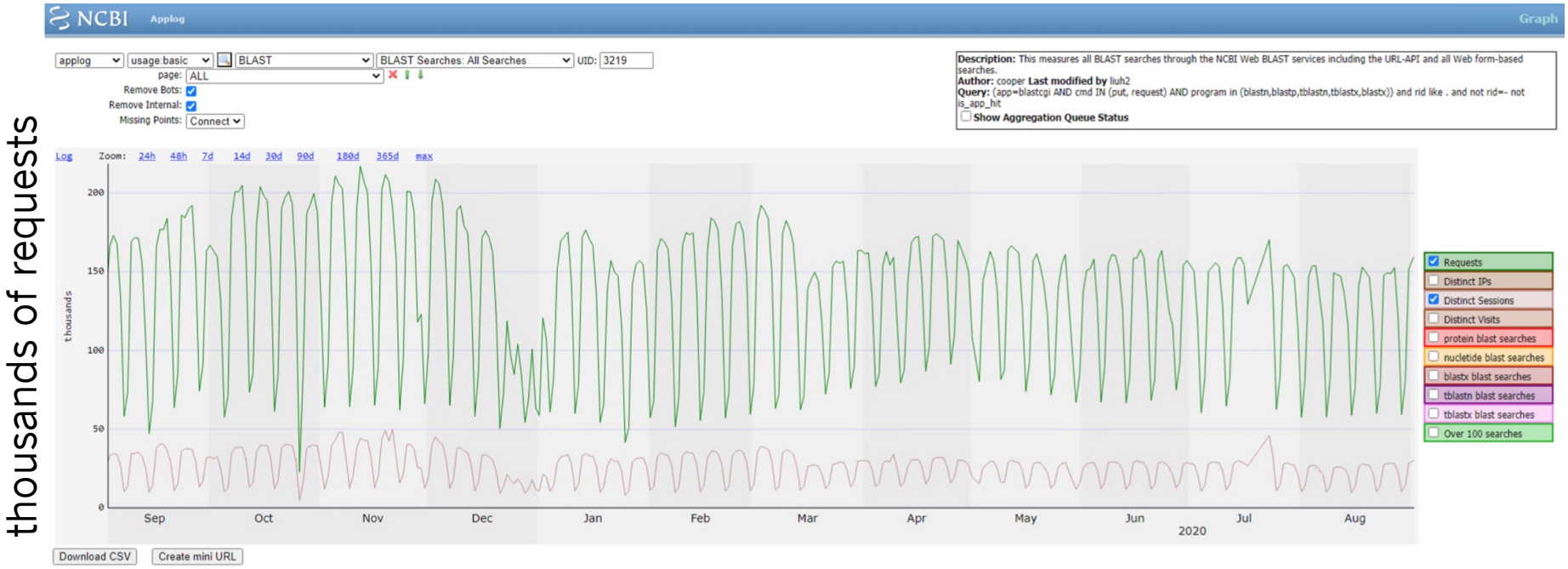
Bowtie is an ultrafast, memory-efficient short read aligner. It aligns short DNA sequences (reads) to the human genome at a rate of over 25 million 35-bp reads per hour. Bowtie indexes the genome with a Burrows-Wheeler index to keep its memory footprint small: typically about 2.2 GB for the human genome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

OSI certified

Bowtie: short-read alignment software

More interconnected methods

NCBI: 150k+ BLAST requests each weekday

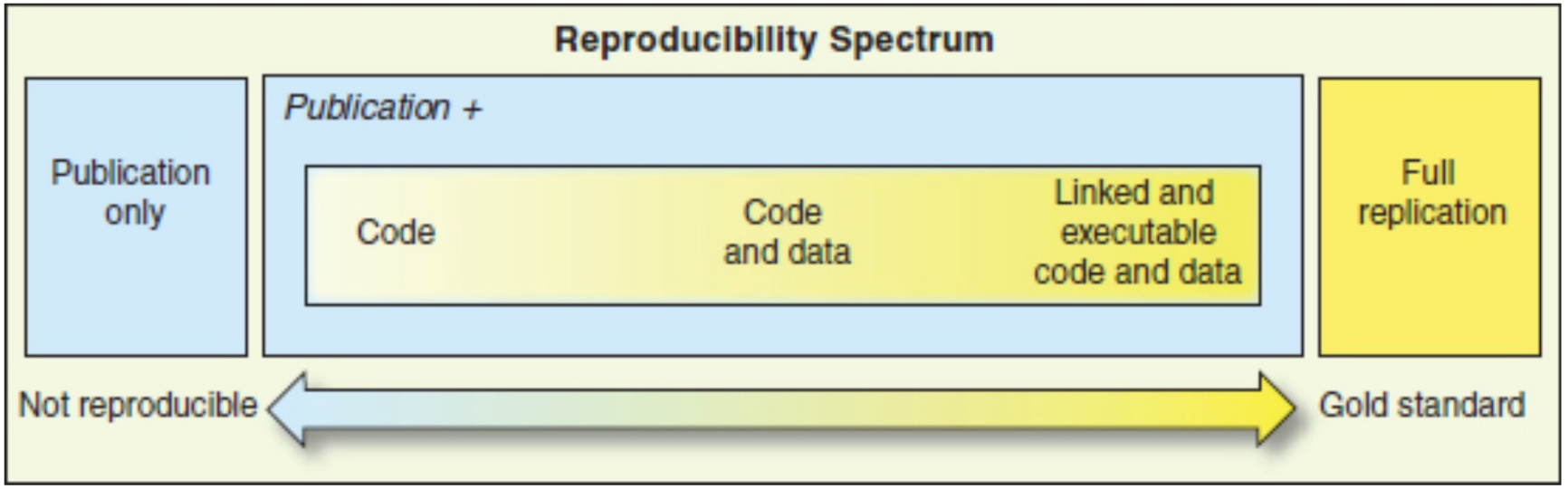


thousands of requests

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
- cellular biology
- epigenetics
- cancer biology
- mutation and evolution
- drug development
- immunology
- microbiome

Why are *you* interested in bioinformatics?

You wrote:

- computational problem solving
- connect biology + quantitative training
- learn new tools and techniques
- develop confidence
- appreciate history of field
- preparation for future career
- build an analysis pipeline
- pipetting hurts your thumb

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-Biol4220/home

Contains links to:

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

Labs

Each lab focuses on a new skill, but may require the use of previously learned skills

One lab is assigned per meeting (due after 1 week)

Completed alone or in groups, but each student must turn in their own work

Labs will be submitted using an online tool called GitHub Classrooms (introduced in Lab 01)

Course project

Design a pipeline to analyze genes

Pipelines will build upon lab skills
- 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)

Exams

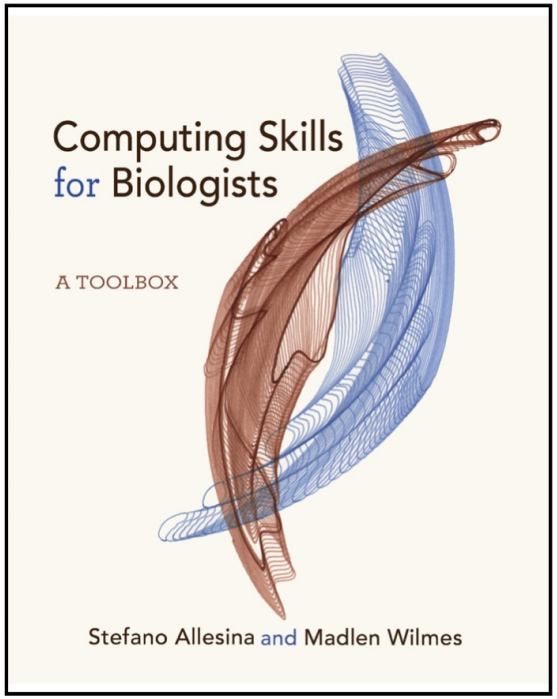
Exam 1 focuses on Weeks 1-7
Exam 2 focuses on Weeks 8-14

Problems closely follow problems from lectures, labs, and reading

Dates are listed on schedule

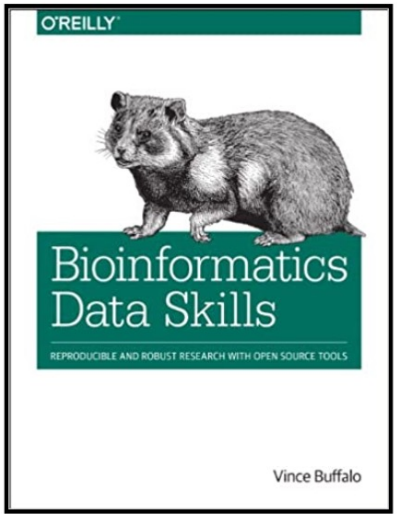
Reading

Primary text

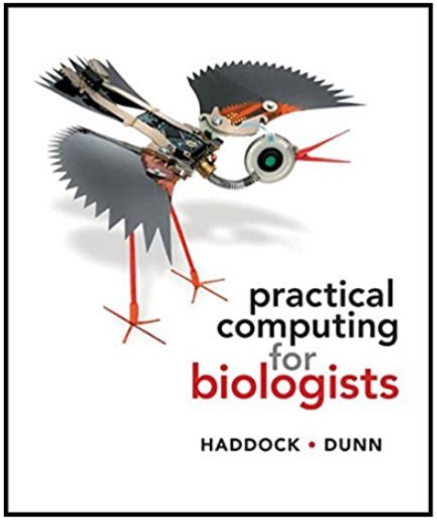


Allesina & Wilmes
ISBN: 9780691167299
~$35

...other useful texts



Buffalo
ISBN: 1449367372
~$35



Haddock & Dunn
ISBN: 0878933913
~$60

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)

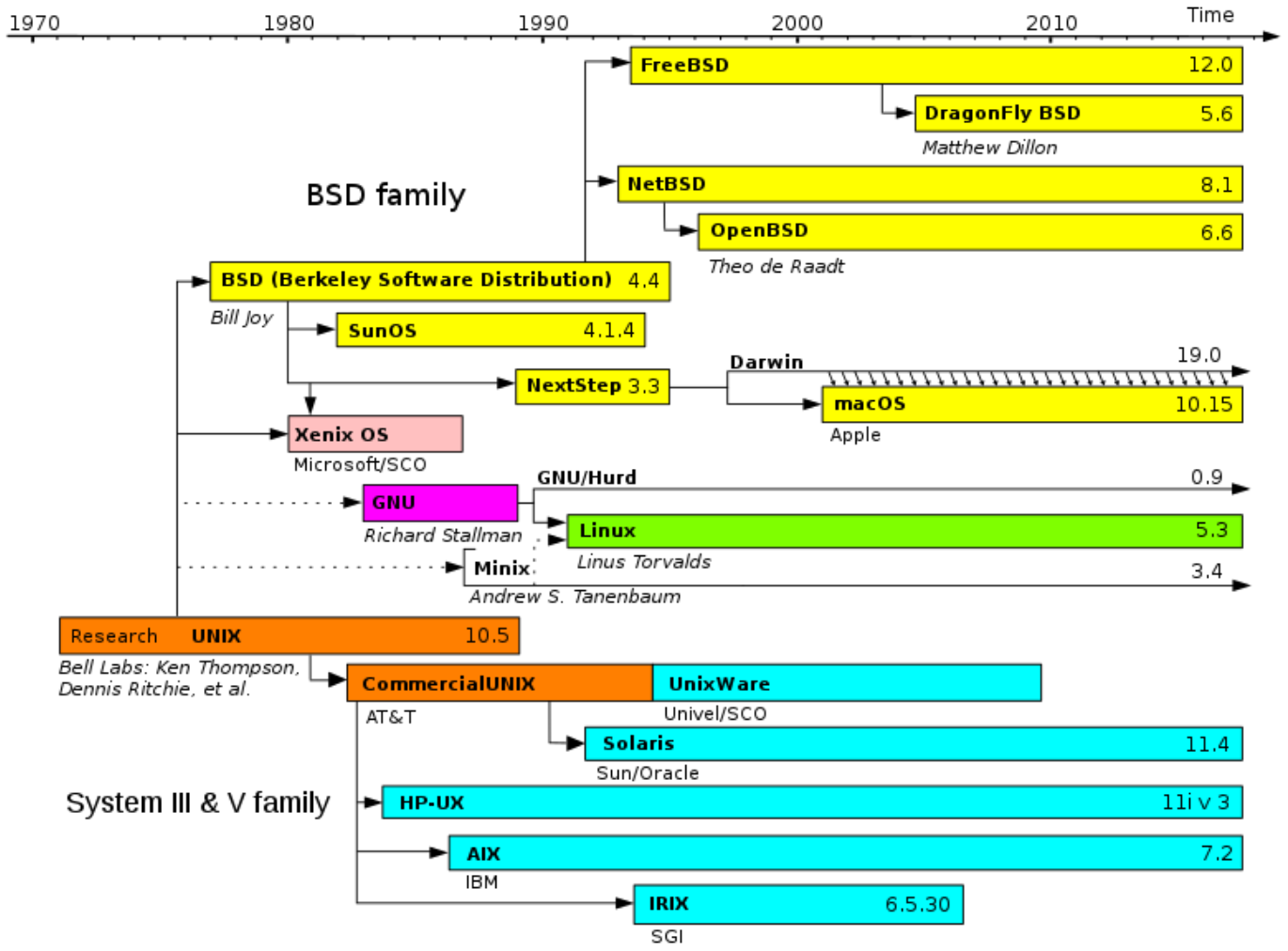
Most scientific computing uses ***Unix***-based systems; we'll be using the Linux distribution, ***Ubuntu***

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

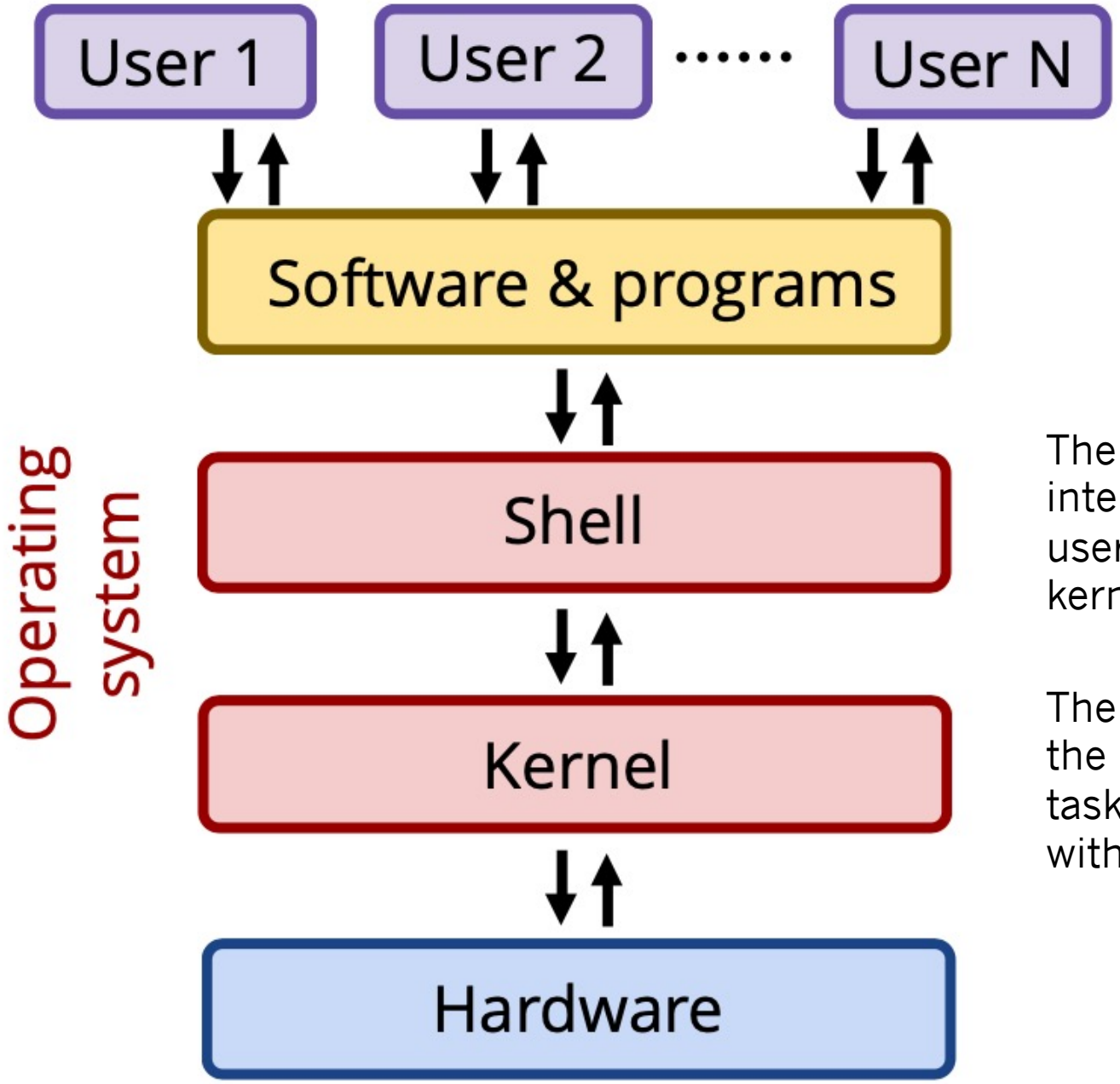


https://commons.wikimedia.org/wiki/File:Unix_timeline.en.svg

Ubuntu

We'll use Ubuntu 20.04 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

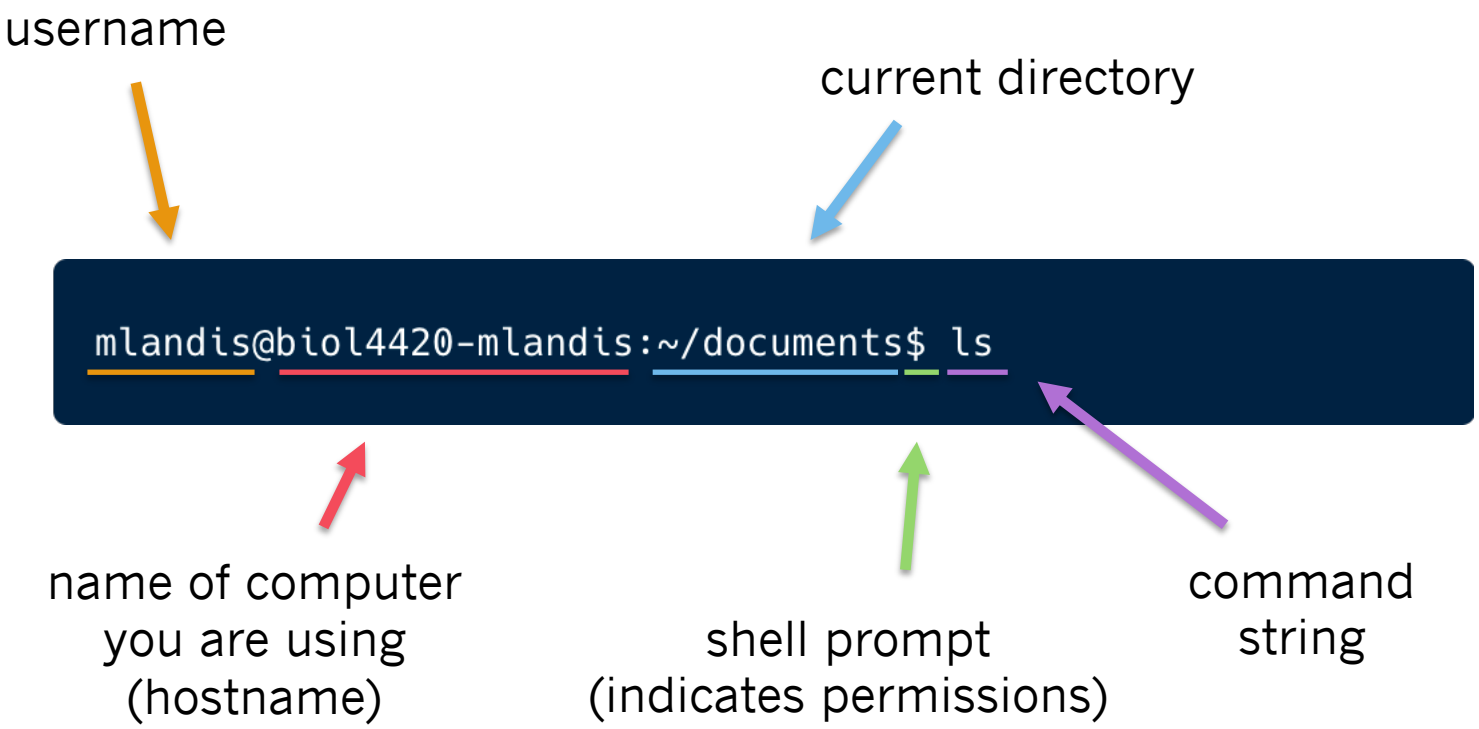
The ***kernel*** has control over all computer resources, including processors, memory, storage, devices, task management, etc.

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
```

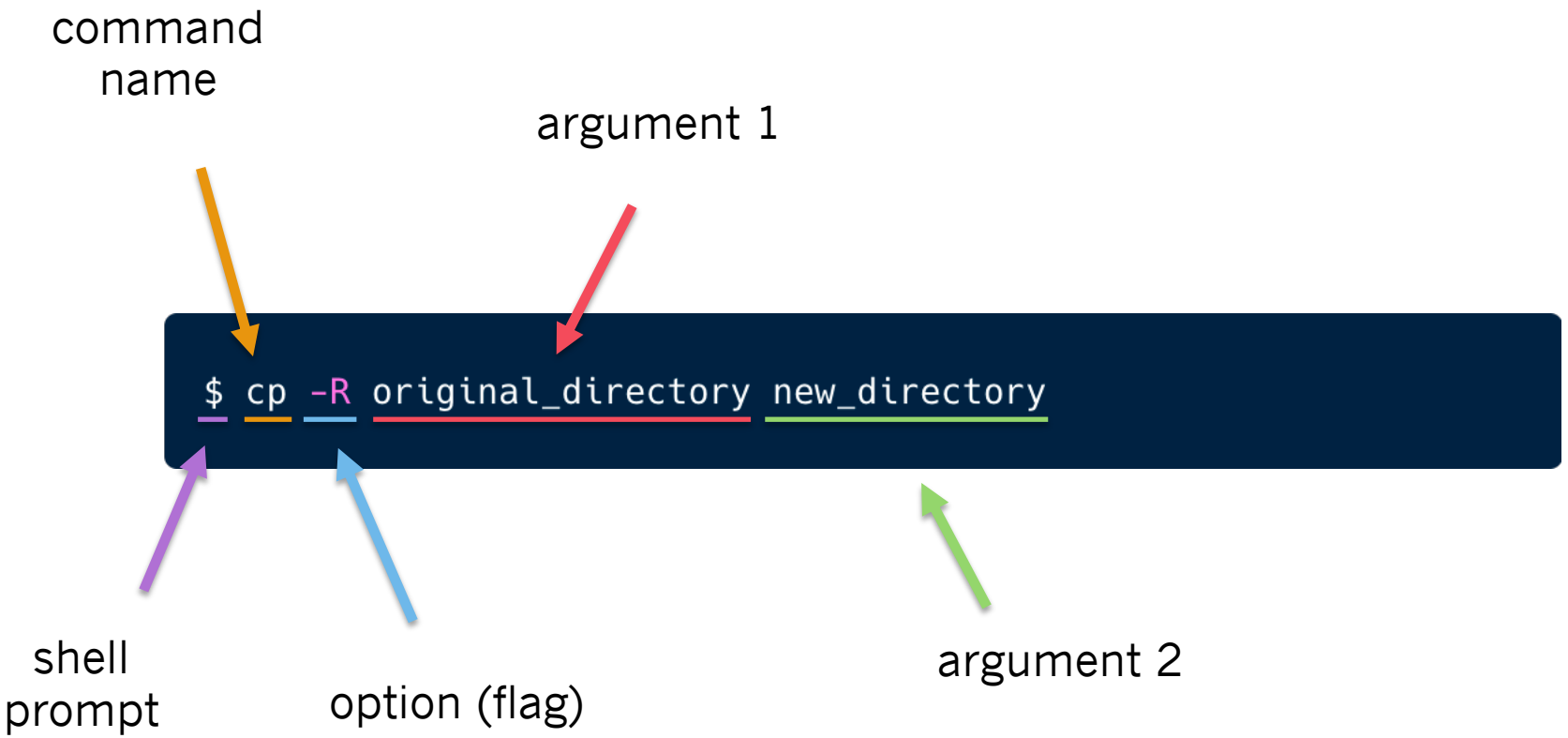
Example of Unix shell command

Command line



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

Command string

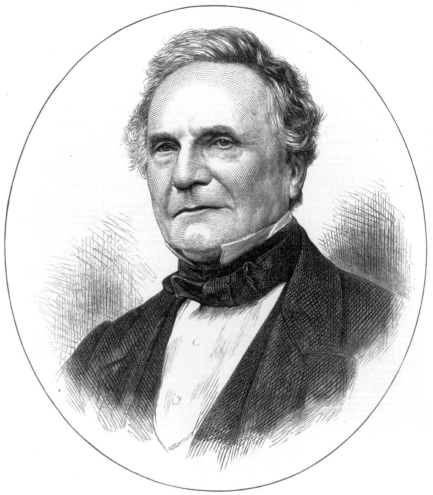


A ***command*** is applied against an ***argument(s)*** and its behavior can be modified by ***option(s)***

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