CMSE 890:301
Overview of Course and Programming

A. Black P.
2018.01.08

Bioinformatics

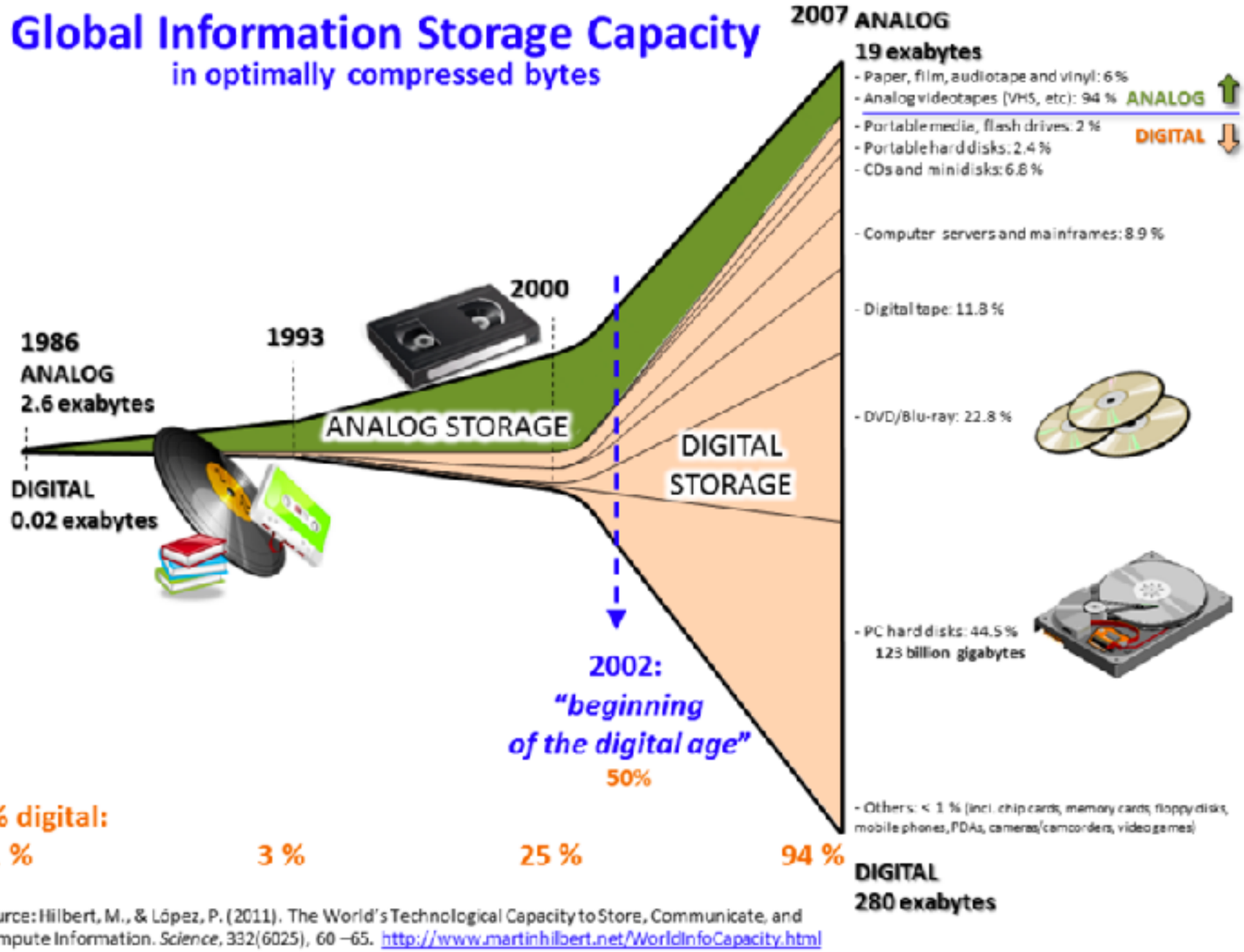
Bioinformatics /ˌbaɪ.oʊˌɪnfərˈmætɪks/ (listen) is an interdisciplinary field that develops methods and software tools for understanding biological data. As an interdisciplinary field of science, bioinformatics combines computer science, statistics, mathematics, and engineering to analyze and interpret biological data. Bioinformatics has been used for *in silico* analyses of biological queries using mathematical and statistical techniques.

Wikipedia

- No clear definition: take biology, chemistry, computer science, and mathematics, and mash them together

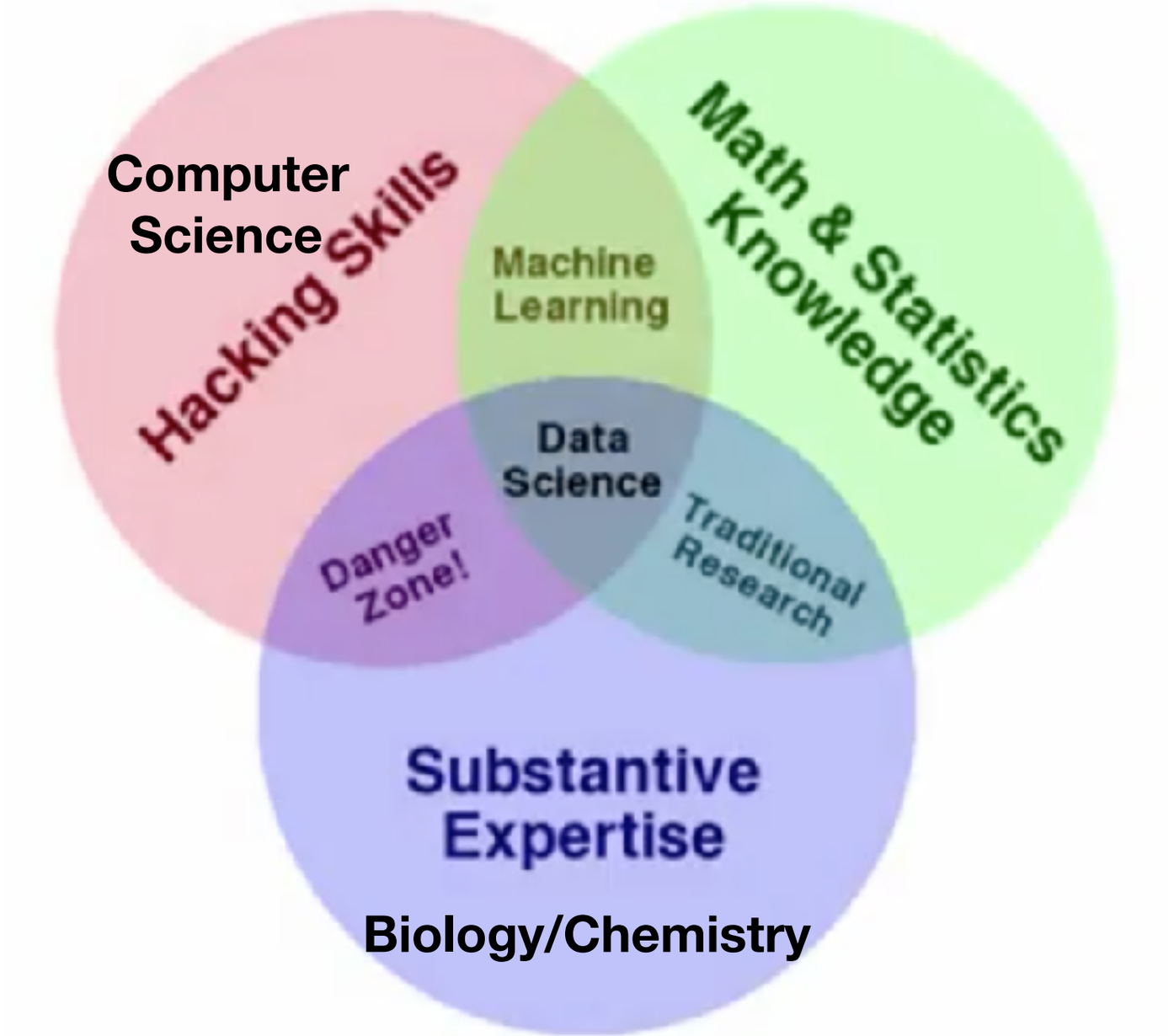
- Diverse fields within bioinformatics

Big Data



Source: Hilbert, M., & López, P. (2011). The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025), 60 –65. http://www.martinhilbert.net/WorldInfoCapacity.html

Data Science and Bioinformatics

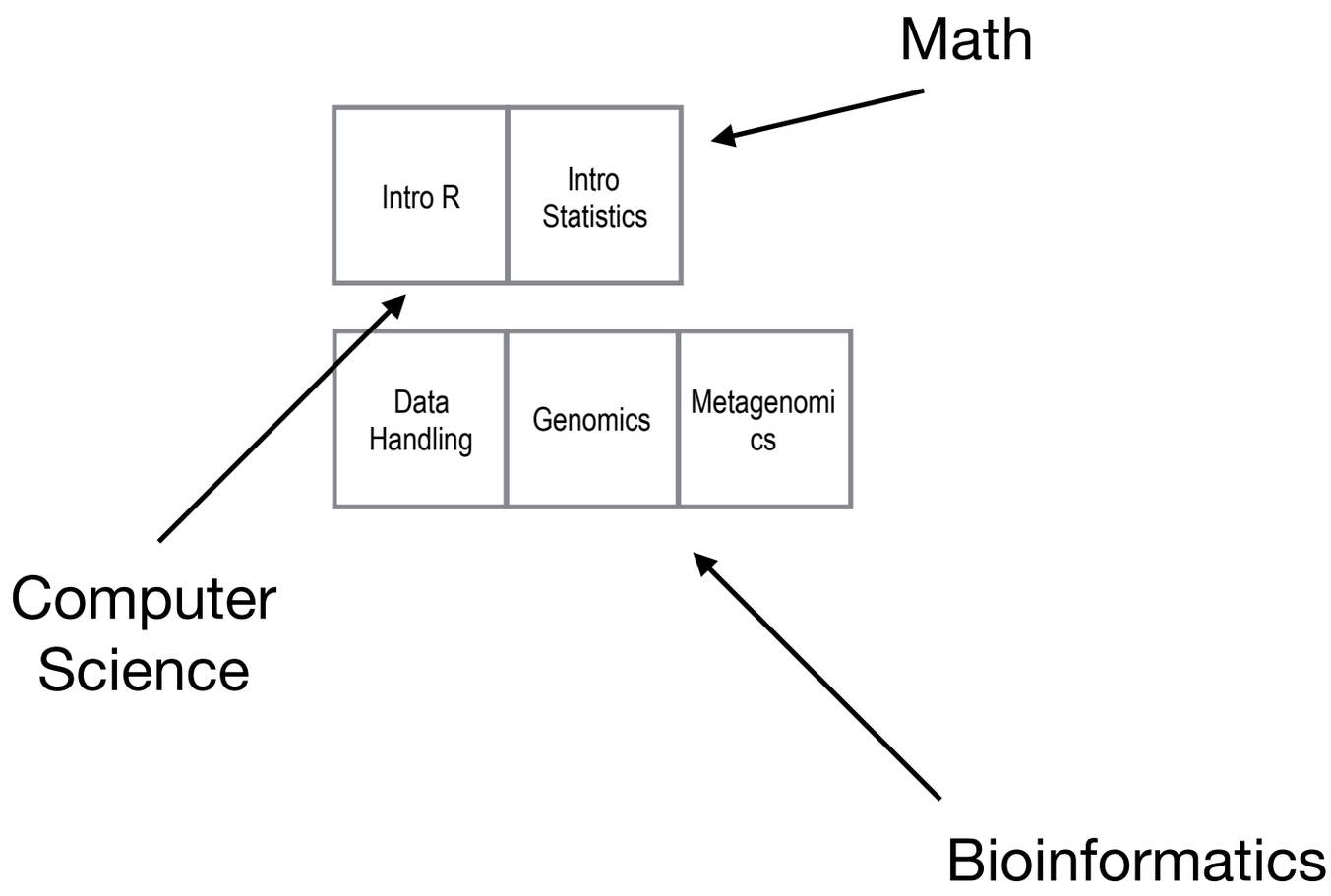


Drew Conway

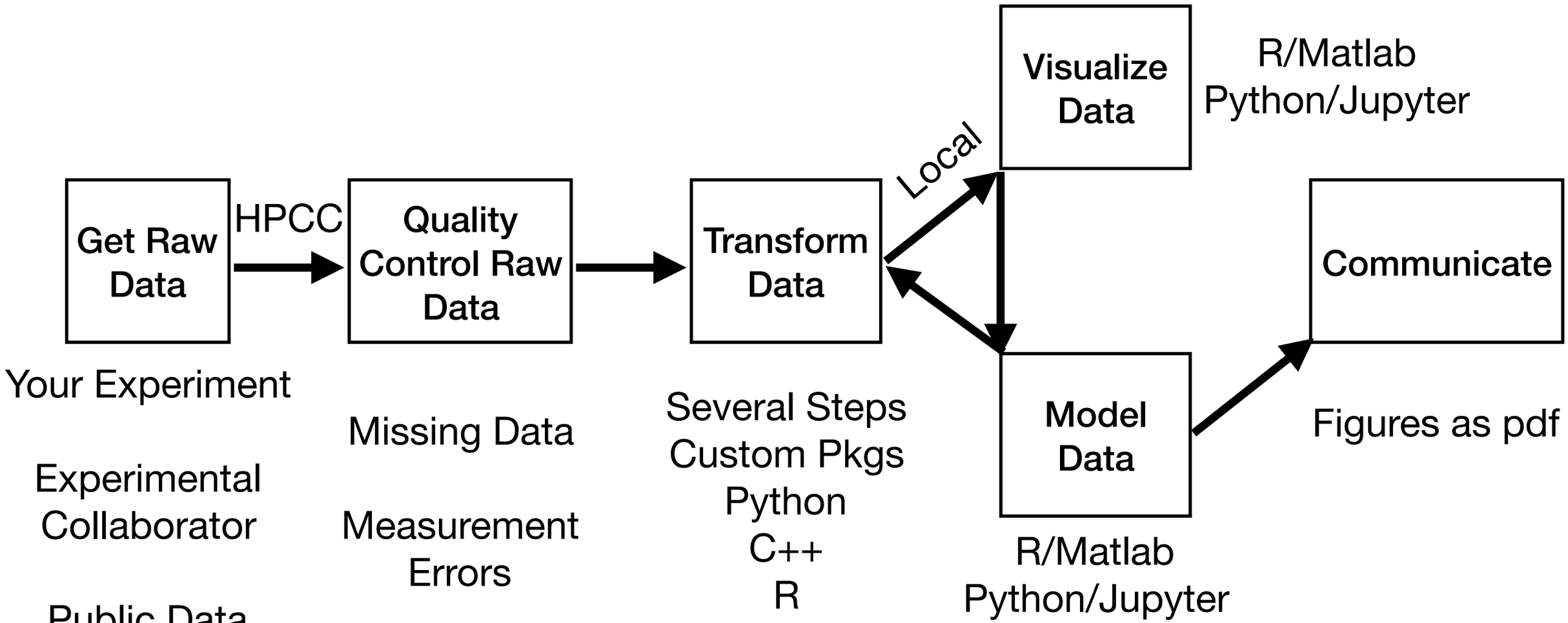
Bioinformatics Skill Development

- Year 1: Nothing works
- Year 2: Student can run data through standard tools (Tool Joy!), but results are unbelievable
- Year 3: Student questions the tools and skills because data are unbelievable, but doesn’t know what to do
- Year 4: Student makes new tools, but new tools are hard to build
- Year 5: Student has new tools and data that makes sense and is rushing to finish

Modules this Semester



Bioinformatics Workflow



Programming

- Writing a series of instructions for the computer to use to complete tasks

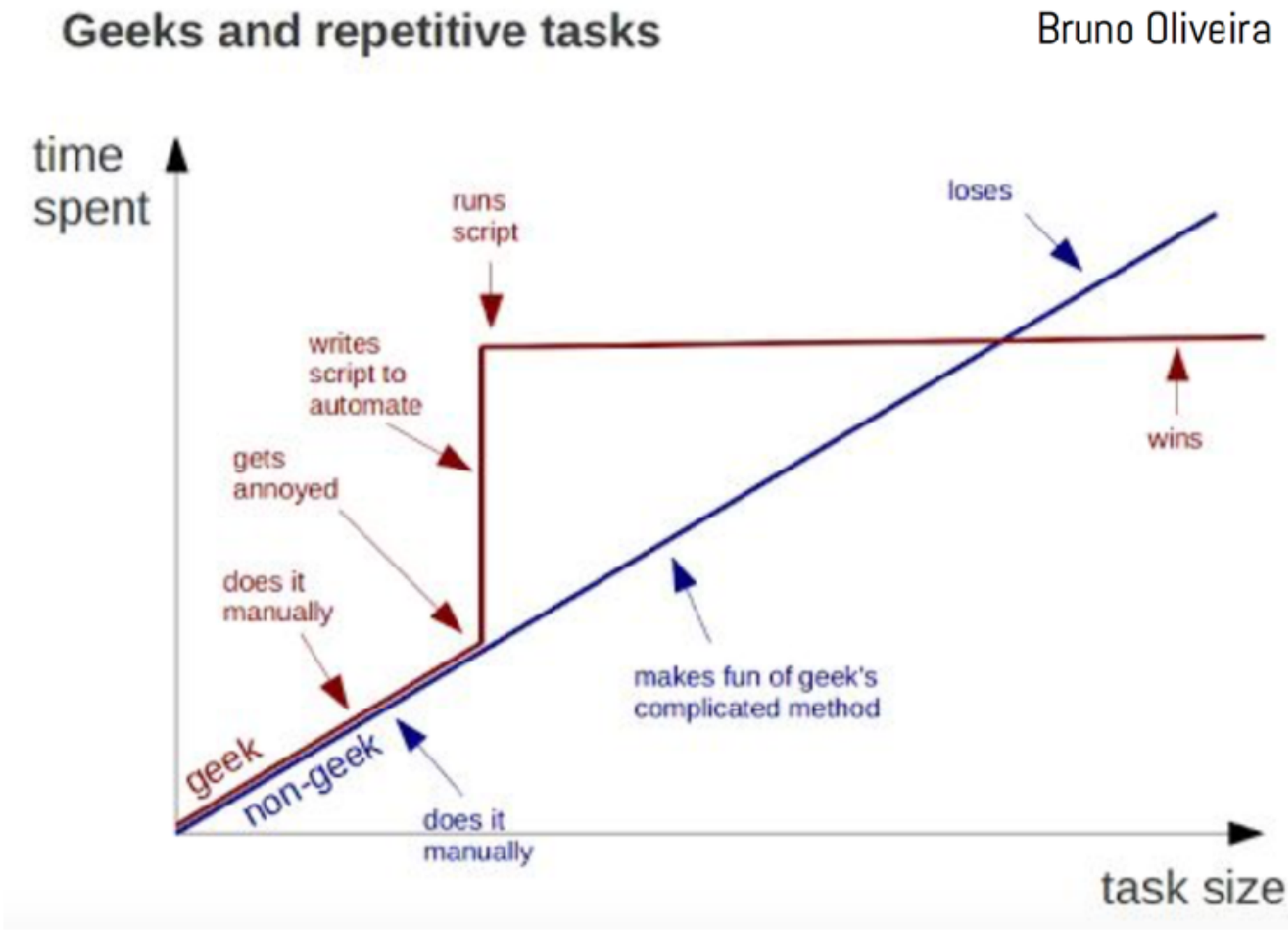
- Steps must be in logical order

- Computer cannot intuit your instruction

- Automate repetitive tasks (computers are good at them, humans are not)

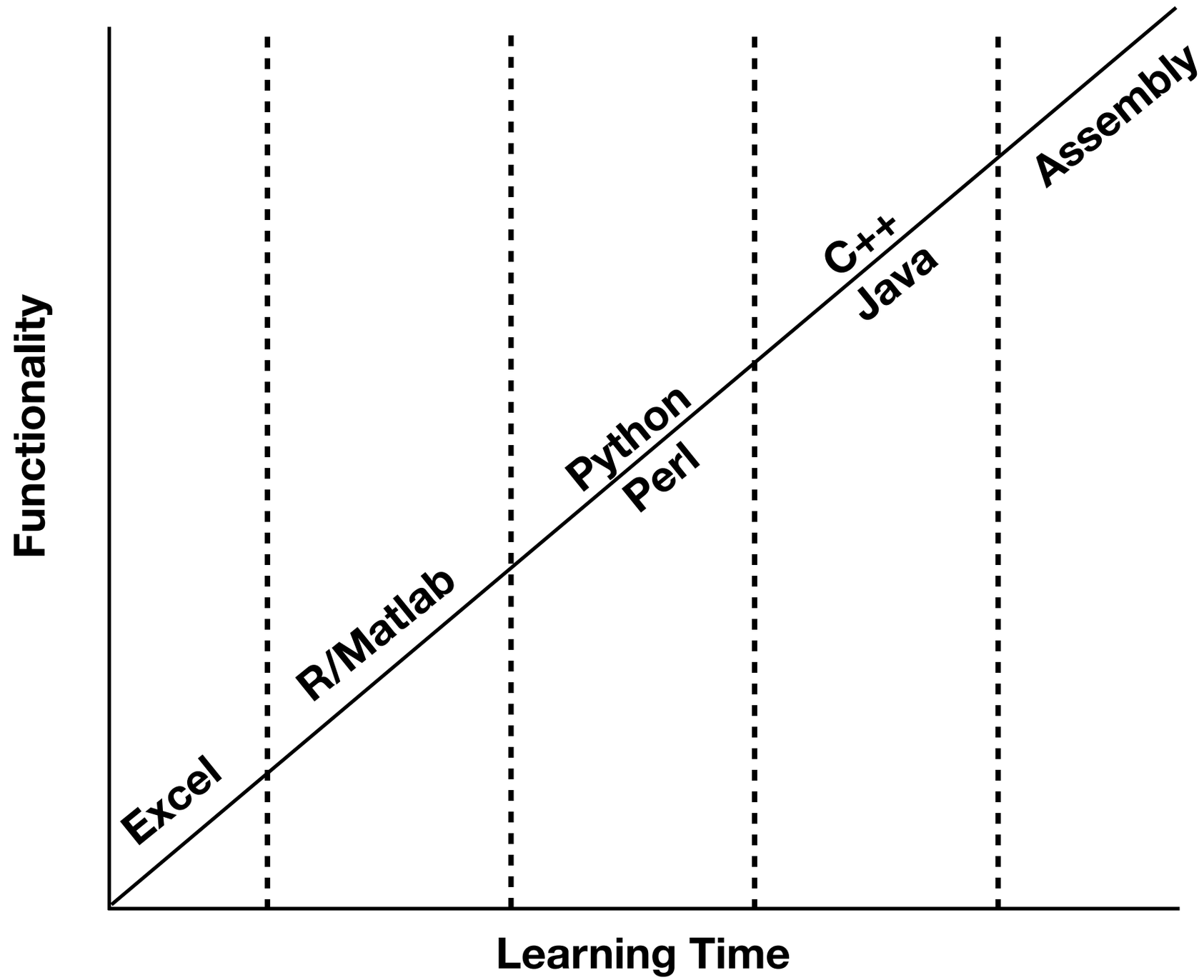
- Running scripts is reproducible; processing data by hand is not

Why Program?


Geeks and repetitive tasks
Bruno Oliveira
time spent
runs script
loses
writes script to automate
wins
gets annoyed
does it manually
makes fun of geek's complicated method
geek
non-geek
does it manually
task size

Languages

- Compiled
  - C/C++
  - Java
- Interpreted
  - Python
  - Perl
  - Ruby
  - Javascript
  - Tcl
  - R



Which Language?

- How hardware/memory intensive is the calculation?
- Are you calculating, data handling, or doing statistics?
- Do you need a graphical user interface?
- What packages/libraries exist for your area?

History of R

- Original language was called “S” and was written by John Chambers at Bell Labs in 1960-70s.

- “R” was developed as an implementation of "S" and named for Ross Ihaka and Robert Gentleman. Written in 1992-2000.

- Most noted for statistics implementation. Still useful because the platform interfaces with many packages and other languages.

Why use R?

- Free licensing
- Get up and running/programming quickly
- Nice graphical user interfaces available for all platforms
- Automate data handling
- Access tons of bioinformatics packages
- Avoid nitpicky hardware details like memory allocation
- Do statistics
- Make publication-quality figures

Issues with R

- Syntax and data structures can be different than many modern programming languages
- Slooooooow
- Hard to scale on High Performance Computing Clusters
- Dependent on open-source software

Getting Help

- Published books (O'Reilly)
- Language website (R-project)
- Online tutorials
- Forums (stackoverflow, biostars, seqanswers)

Integrated Design Environments