

## Definitions (dictionary.com)

- **bioinformatics** *n.*
  - the collection, classification, storage, and analysis of biochemical and biological information using computers especially as applied in molecular genetics and genomics
- **genomics** *n.*
  - The study of all of the nucleotide sequences, including structural genes, regulatory sequences, and noncoding DNA segments, in the chromosomes of an organism.
  - a branch of biotechnology concerned with applying the techniques of genetics and molecular biology to the genetic mapping and DNA sequencing of sets of genes or the complete genomes of selected organisms using high-speed methods, with organizing the results in databases, and with applications of the data (as in medicine or biology) – compare PROTEOMICS

## The role of programming

- **How does programming relate to**
  - Bioinformatics
  - Genomics
  - Computer Science
- **How does programming relate to**
  - Development and analysis of algorithms
  - Engineering (Software and Otherwise)
  - Artistic endeavors
- **Is Dijkstra relevant?**

## Edsger Dijkstra [http://en.wikiquote.org/wiki/Edsger\\_Dijkstra](http://en.wikiquote.org/wiki/Edsger_Dijkstra)

- "Computer Science is no more about computers than astronomy is about telescopes."
- "I mean, if 10 years from now, when you are doing something quick and dirty, you suddenly visualize that I am looking over your shoulders and say to yourself: 'Dijkstra would not have liked this', well that would be enough immortality for me."
- "Object-oriented programming is an exceptionally bad idea which could only have originated in California."



## Why do both perl and Java exist?

- **Analyze perl programs from Lesk, Bioinformatics**
  - "Perl is designed to make the easy jobs easy and the hard jobs possible." -Larry Wall.
  - What is good about these programs
  - What is bad about these programs
  - How can we engineer them to be better?
- **What is Java about?** <http://java.sun.com/docs/overviews/java/java-overview-1.html>
  - A simple, object-oriented, network-savvy, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, dynamic language.

## Algorithms

- **Why is the alignment problem hard?**
  - How do we do spell-checking?
  - How do we make change?
  - How does Google adjust our queries?
- **When does fast matter**
  - Always?
  - 80/20 rules?
- **Relationship to Computer Science and Programming**
  - Back to Dijkstra or the future?