

# Algorithms on Strings, Trees, and Sequences

Introduction

*by Marina Barsky*

# Sequences (Strings)

*Strings* are natural groupings of symbols into sequences, where the order has a special significance

bad salad

≠

sad ballad



Symbols: a b d | s

# The course is about:

- String Algorithms (pattern matching, indexing, grouping, prediction): ideas, pseudocode, complexity
- Strings of interest are long and not broken into tokens (words)
- Trees: derived from strings

The area is called ***Stringology***

# The goal:

- To become familiar with the **problems** of modern Stringology
- To be able to identify which of these problems are efficiently **computable**
- Acquire **algorithmic tools** to solve these problems

## Required Background:

- Algorithms
- Data structures
- Probability

# Deliverables

- Assignments – 40 %
- Class work\* – 30 %
- Final Project \*\* – 30 %

---

\* Consists of:

- In-class quizzes – to monitor comprehension
- In-class activities – to learn how to communicate your ideas

\*\* Term paper or implementation

# Long strings

## Some texts

不貨心其知無不其  
 尚使不腹無為盈紛  
 賢民亂弱欲則淵和  
 使不是其使無兮其  
 民為以志夫不似光  
 不盜聖強智治萬同  
 爭不人其者道物其  
 不見之骨不冲之塵  
 貴可治常敢而宗湛  
 難欲其使為用挫兮  
 得使心民也之其似  
 之民實無為或解或

Tao Te Ching by Lao Tzu

## Music scales



Saint-Saëns, Camille (1835-1921), Carnaval des Animaux, Orch. & 2 Pfts., Aquarium



Beethoven, Ludwig Van (1770-1827), Für Elise, Pft.

ED#ED#EAED#EFDC#DECHCDH (S-S)

ED#ED#EHDACEAHEG#C (B)

## Information Retrieval

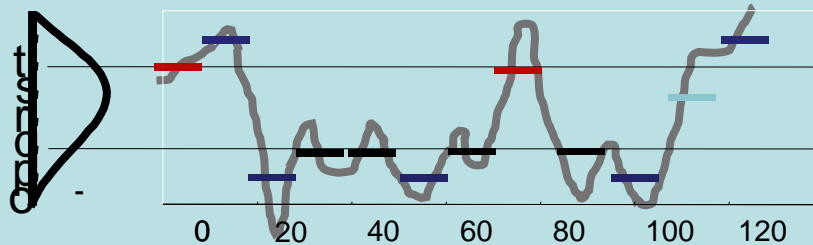
Collection of documents:

- mouse eats cheese
- cat eats mouse
- snake eats mouse

Query: who eats mouse?

Strings of words

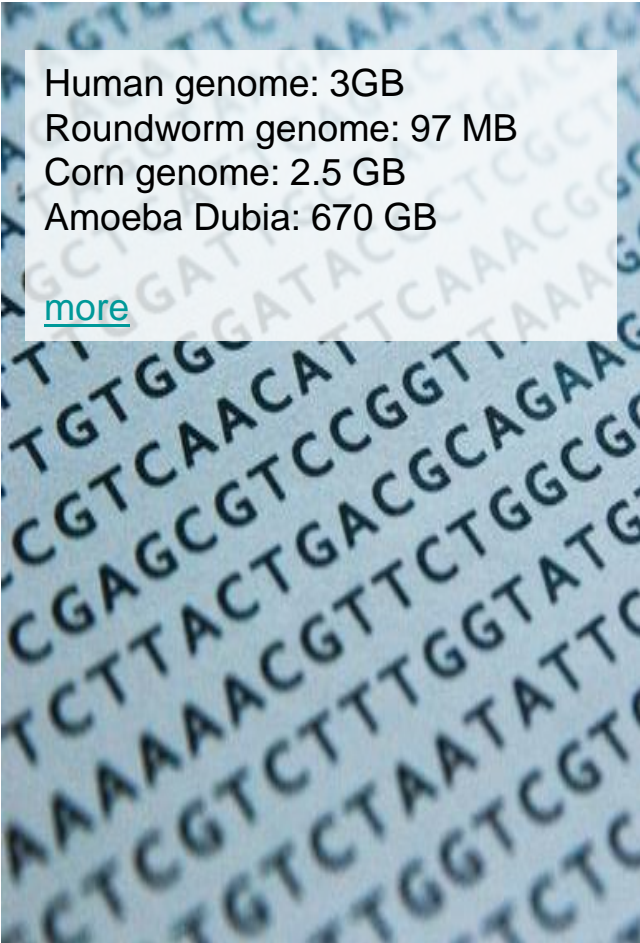
## Time series



stoppopsport

SAX - Symbolic Aggregate approximation (by Eamon Keough, 2001)

# Very long strings



Human genome: 3GB  
Roundworm genome: 97 MB  
Corn genome: 2.5 GB  
Amoeba Dubia: 670 GB

[more](#)

Sequences of molecules in different biological polymers:

*DNA, RNA, proteins*

Digitalization of the molecular code  
→ new type of data:

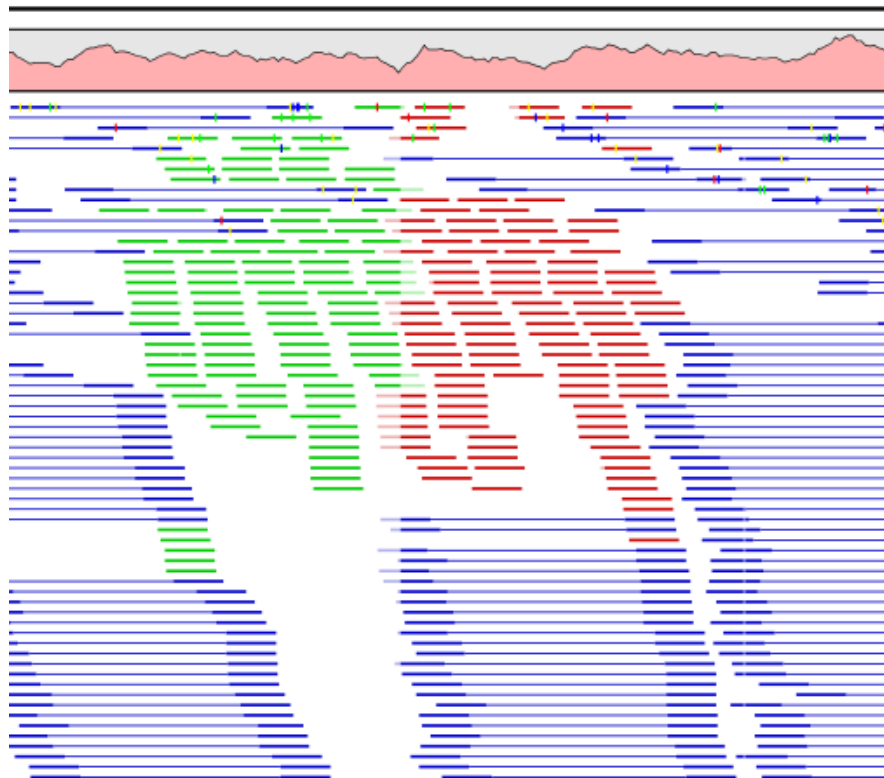
- no clear separation into tokens
- large token size (30,000 and more)
- long “texts” (247 MB in chromosome I)
- virtually unlimited number of different substrings ( $2 \times 10^{17}$  in Human genome)

Experiment with large strings:

[https://barsky.ca/marina/UTOR/experiments/bio\\_example/index.html](https://barsky.ca/marina/UTOR/experiments/bio_example/index.html)

## Input Dataset:

20 **TB** of short DNA reads from 232 individuals





# Molecular Biology:

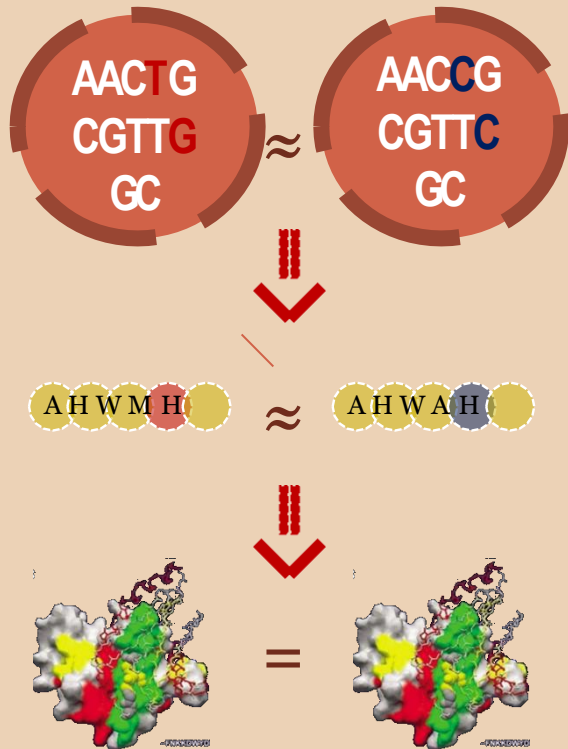
- Describes living things in terms of chemical matter (molecules) and chemical and physical mechanisms
- Studies macromolecules – DNA, RNA, protein – and the mechanisms of their interaction

# Bioinformatics:

- Applies concepts of Informatics and Computer Science to the field of Molecular Biology – to extract new knowledge from the information embedded in genetic code

# Bioinformatics: assumptions

**Similar sequences –  
similar function**



If function is unknown – look at similar sequences with known functions

**Similar function – partly  
similar sequences**



If same function – look at similar substrings which may be responsible for it

# Protocol of converting a biological problem into CS problem

1. Biological question (find *similar* sequences)
2. Formalization (how to measure *similarity*)
3. Design an *efficient* algorithm to solve the *formalized* problem
4. Model + learning – learn parameters of an algorithm from real data
5. Evaluation of results – distinguish (statistically) significant results from artifacts
6. Presentation of the results

# Example 1: find similar sequence

- Input:
  - Query: sequence of DNA bases:  
**AACCCTTAG**
  - The set of sequences of known genes (with their functions):  
**ACCTAG**  
**AGCCCGTA**  
**AAGCCGCTTA**
- Question: **which one** is the most similar to the **query sequence**?

# Which pair is most similar?

1

AACCCTTAG

ACCTAG

2

AACCCTTAG

AGCCCGTA

3

AACCCTTAG

AAGCCGCTTA

# Example 2: evolutionary tree

- Input: four DNA sequences taken from four species.



AAG



AAA



AGA



GGA

# Following protocol

1. Biological question: which evolutionary tree *best* explains these sequences ?
2. Formalization: what is the metric for *the best* tree?

Let it be *the parsimony principle*

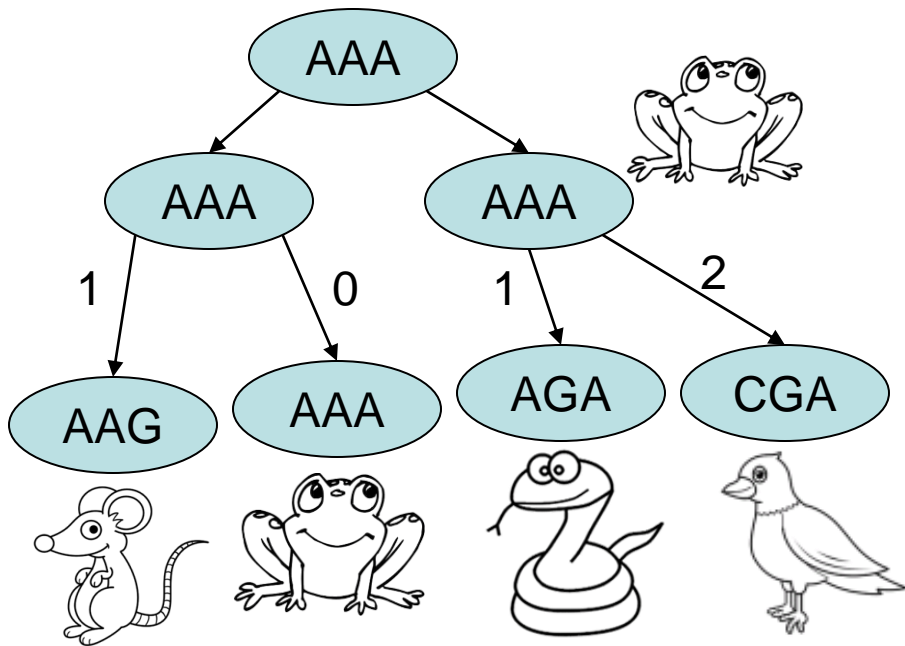


# Parsimony principle

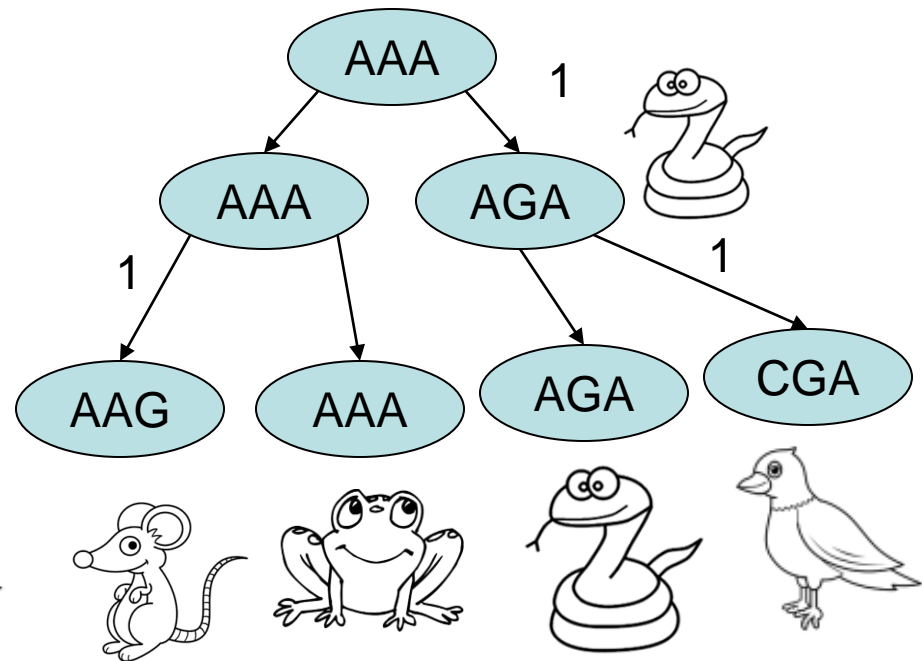
- In science, ***parsimony*** is preference for the least complex explanation. This is regarded as good when judging hypotheses.
- Occam's razor also states the "principle of parsimony": *entia non sunt multiplicanda praeter necessitatem*, is the principle that "entities must not be multiplied beyond necessity": **the simplest explanation or strategy tends to be the best one**
- Under maximum parsimony, **the preferred phylogenetic tree is the tree that requires the smallest number of evolutionary changes.**

# Many possible trees

Tree 1



Tree 2



What tree is more parsimonious?

# Next steps

3. Efficient algorithm: how can we compute the best tree efficiently?
  4. Adjusting parameters from the data: A is more likely to be replaced by G or by T?
  5. Significance: is the best tree found significantly (statistically) better than others ?
  6. Present results as a tree
- The main question remains: does the tree make biological sense ?

# We will discuss solutions to the following sample problems

- Sequence comparison
- Pattern discovery
- Gene finding
- Sequence-based evolution

# Algorithmic Tools: outline

- Discrete algorithms:
  - Combinatorial pattern matching
  - String indexing
  - Dynamic programming
- Probabilistic models:
  - Hidden Markov Models
  - Maximum likelihood
  - Bayesian inference
- Hard problems:
  - Heuristics
  - Approximation algorithms

# 'Strings' of life

- DNA
- RNA
- Proteins