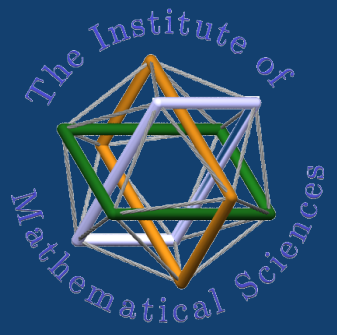
Sequence Analysis:

Bits & Scripts II
March 13-24, 2025

Md. Izhar Ashraf
Computational Epigraphy Lab(iCEL),
The Institute of Mathematical
Sciences(IMSc), Chennai
ashraf@imsc.res.in



Sequence Analysis in Biology: Examples

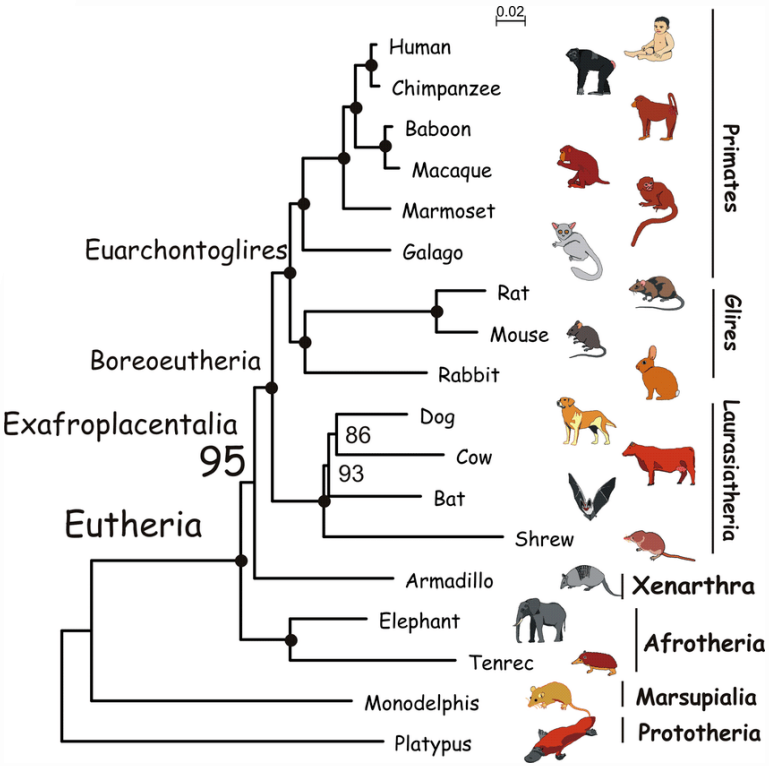


Figure: Source: PLoS genetics 2007

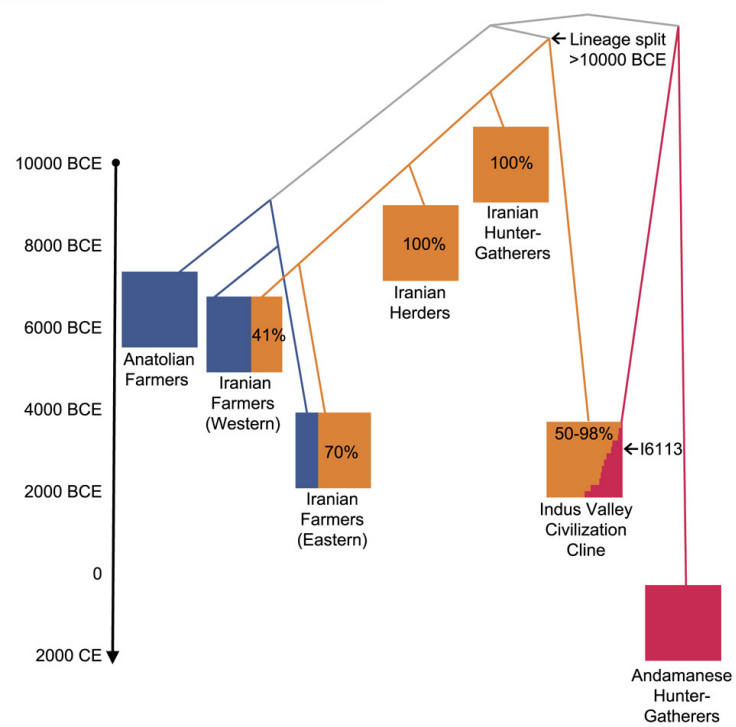


Figure: Cell 2019

Sequence Analysis in Economics: Examples

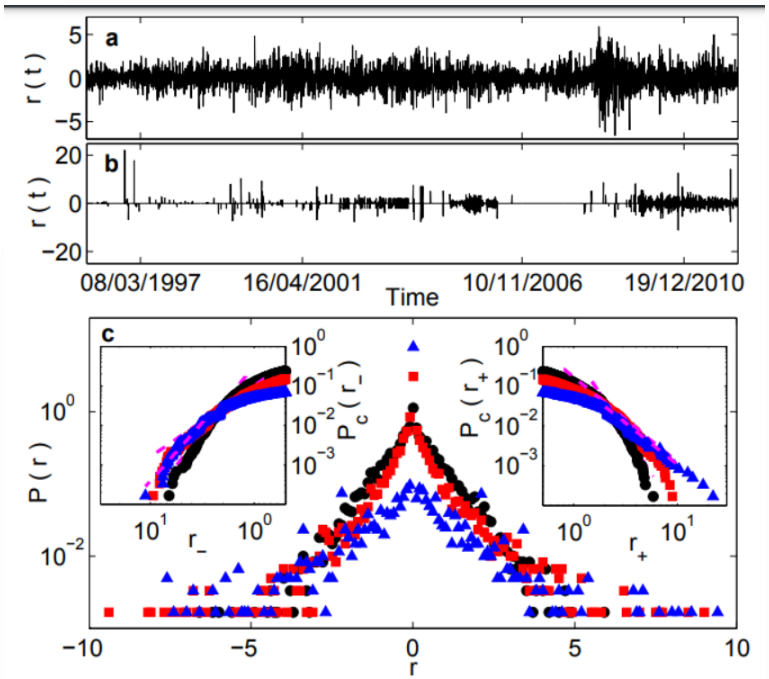


Figure: Source: Sitabhra et. al Physica A 2018

Figure: Source: Tradingview.com

Sequence Analysis in War, Business and Politics

Figure: Source: Wikipedia

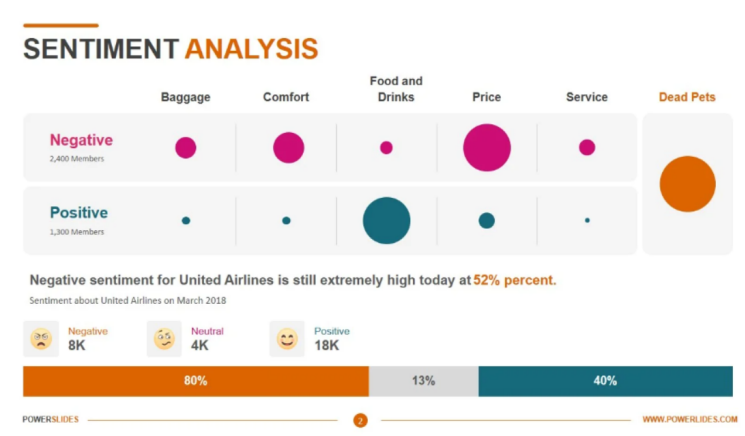


Figure: Source: Unknown

Sequence Analysis in Archaeology

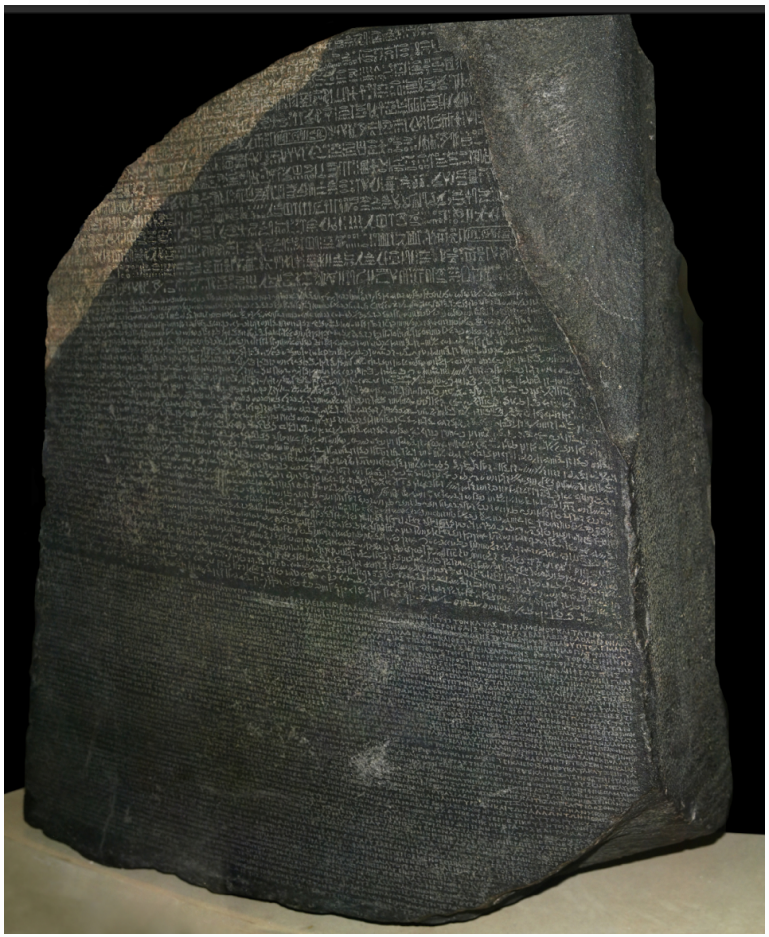


Figure: Source: Wikipedia

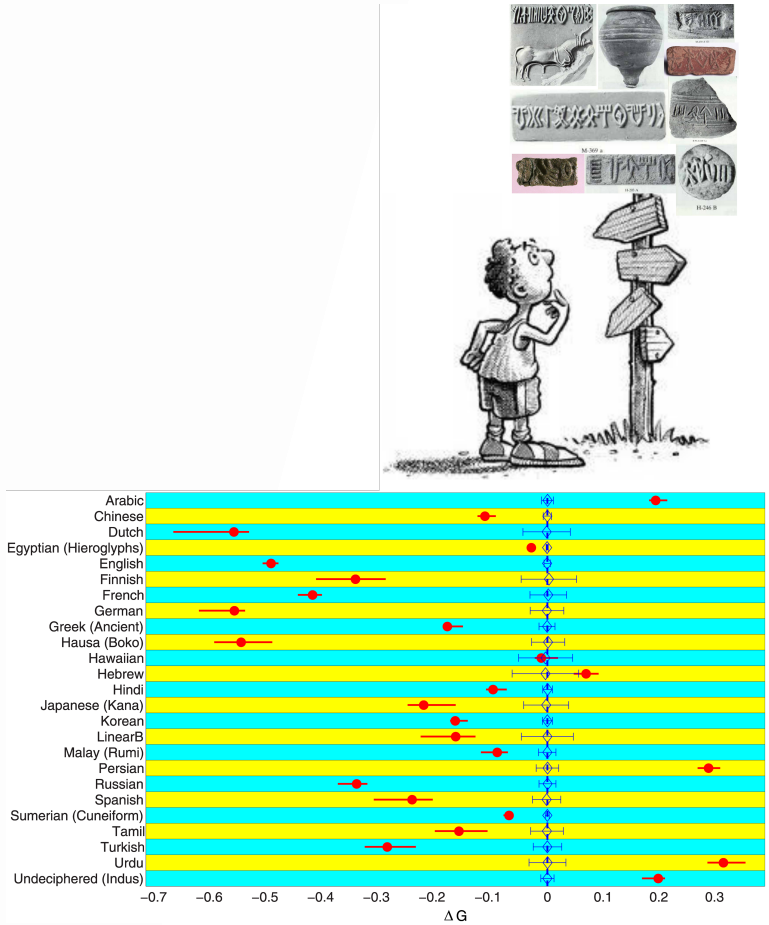
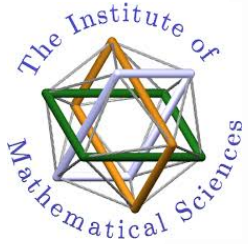


Figure: Source: In-house

Sequence Analysis in Games

Figure: Source: nytimes.com



What is Sequence Analysis?

- Sequence analysis is the process of examining and comparing sequences of symbols.
- These sequences can represent:
  - Biological data (DNA, RNA, proteins)
  - Textual data (words, characters)
  - Any data that can be represented as a linear sequence.
- The goal is to identify patterns, similarities, and differences that reveal underlying relationships.
- Applications:
  - Bioinformatics (evolutionary studies, gene identification)
  - Natural Language Processing (text similarity, error correction)
  - Historical text analysis (Cipher breaking, text comparison)

Global Alignment: Needleman-Wunsch

- **Objective**: Optimal end-to-end sequence alignment.
- **Formula**:

Initialization: $F(i,0) = -i \cdot d, \quad F(0,j) = -j \cdot d$

Recurrence: $$F(i,j) = \max \begin{cases} F(i-1,j-1) + s(x_i, y_j) \\ F(i-1,j) - d \\ F(i,j-1) - d \end{cases}$$

- **Example**: Align $X = \text{GAT}$, $Y = \text{CAT}$ ($s_{\text{match}} = +1$, $s_{\text{mismatch}} = -1$, $d = 1$):

	$-$	C	A	T
$-$	0	-1	-2	-3
G	-1	-1	-2	-3
A	-2	-2	0	-1
T	-3	-3	-1	1

Final Alignment:

$$\begin{array}{ccc} G & A & T \\ | & | & | \\ C & A & T \end{array}$$

Score $= 1$

- **Applications**: Reconstructing fragmented texts, aligning full gene sequences.

Local Alignment: Smith-Waterman

- **Objective**: Identify high-similarity subregions.
- **Formula**:

$$S(i,j) = \max \begin{cases} 0 \\ S(i-1,j-1) + s(a_i, b_j) \\ S(i-1,j) - d \\ S(i,j-1) - d \end{cases}$$

- **Example**: Align $X = \text{GATT}$, $Y = \text{ATT}$ ($s_{\text{match}} = +2$, $d = 2$):

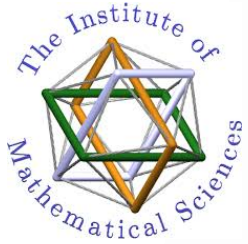
	$-$	A	T	T
$-$	0	0	0	0
G	0	0	0	0
A	0	0	2	0
T	0	0	2	4
T	0	0	0	6

Best Local Alignment:

$$\begin{array}{ccc} A & T & T \\ | & | & | \\ A & T & T \end{array}$$

Score $= 6$

- **Applications**: Detecting conserved motifs, finding deity names in inscriptions.

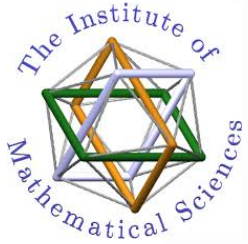


Edit Distance: Levenshtein Distance

- **Purpose**: Minimal edits (insertions, deletions, substitutions).
- **Formula**:

$$\text{lev}(a,b) = \begin{cases} |a| & \text{if } |b| = 0 \\ |b| & \text{if } |a| = 0 \\ \text{lev}(\text{tail}(a), \text{tail}(b)) & \text{if } a[0] = b[0] \\ 1 + \min \begin{cases} \text{lev}(\text{tail}(a), b) \\ \text{lev}(a, \text{tail}(b)) \\ \text{lev}(\text{tail}(a), \text{tail}(b)) \end{cases} & \text{otherwise} \end{cases}$$

- **Application**: Correcting OCR errors, fuzzy searching.



Levenshtein Distance Example

Example: Transform "K!ng" to "King". Substitution = 1, Insertion/Deletion = 1.

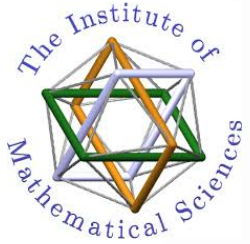
1. **Final Matrix**:

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 1 & 2 & 3 \\ 3 & 2 & 2 & 1 & 2 \\ 4 & 3 & 3 & 2 & \mathbf{1} \end{bmatrix}$$

Alignment:

$$\begin{array}{cccc} K & ! & n & g \\ K & i & n & g \end{array}$$

Total edits = 1 (substitute "!" "i").

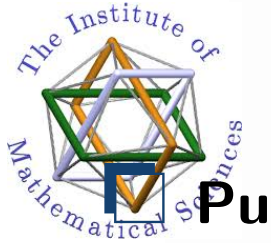


Cosine Similarity: Definition & Formula

- Measures similarity between two vectors by their **orientation or angle** in a multidimensional space
- Range: $[-1, 1]$ (1 = identical, 0 = unrelated, -1 = opposite)
- Applications: Text/document comparison, recommendation systems

$$\text{Similarity} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Key Features: Scale-invariant (ignores vector magnitude)



Jaro-Winkler Distance: Overview

Purpose: Measure string similarity with a bonus for common prefixes. Ideal for short strings (e.g., names, epigraphic terms).

- **Components**:
  - **Jaro Similarity**: Base score based on matching characters and transpositions.
  - **Winkler Adjustment**: Boosts similarity for shared prefixes.

$$J(s,t) = \begin{cases} 0 & \text{if } m = 0, \\ \frac{1}{3}\left(\frac{m}{|s|} + \frac{m}{|t|} + \frac{m-t}{m}\right) & \text{otherwise.} \end{cases}$$

$$JW(s,t) = J(s,t) + l \cdot p \cdot (1 - J(s,t))$$

- **Variables**:
  - m: Matching characters (within window $\left\lfloor \frac{\max(|s|,|t|)}{2} \right\rfloor - 1$)
  - t: Transpositions (half the mismatched order pairs)
  - l: Length of common prefix (max 4)
  - p: Scaling factor (default: 0.1)

Jaro-Winkler Example: MARTHA vs MARHTA

- **Matching Characters**:
  - All 6 characters match (window = 2)

1. **Transpositions**:
   - Swapped T/H: 1 transposition $\Rightarrow t = 0.5$
2. **Jaro Calculation**:

$$J = \frac{1}{3}\left(\frac{6}{6} + \frac{6}{6} + \frac{6 - 0.5}{6}\right) \approx 0.972$$

Matching positions:

M	A	R	T	H	A
|	|	|		|	
M	A	R	H	T	A

- **Matches**: 6/6 characters
- **Transpositions**: 1 pair (T/H)

3. **Winkler Boost** (prefix=3):

$$JW = 0.972 + 3 \cdot 0.1 \cdot (1 - 0.972) \approx 0.980$$

Applications: OCR error correction, deduplication of epigraphic records, name matching in fragmented texts.

Hamming Distance

Definition: Hamming Distance measures the number of positions at which two equal-length strings differ.

$$H(s,t) = \sum_{i=1}^{n} \mathbf{1}(s_i \neq t_i)$$

where:

- s_i and t_i are symbols at position i in sequences s and t.
- $\mathbf{1}(s_i \neq t_i)$ equals 1 if symbols differ, 0 otherwise.

Example: Compare binary strings:

$$s = 110101, \quad t = 100111$$

Differences occur at positions: $2, 4, 5$ **Hamming Distance:** $H(s,t) = 3$

Applications:

- Error Detection and Correction (e.g., ECC, QR Codes)

Caesar Cipher

Definition: A substitution cipher that shifts letters by a fixed number k.

Encryption & Decryption:

$$E(x) = (x + k) \mod 26, \quad D(x) = (x - k) \mod 26$$

Example: Encrypt "HELLO" with $k = 3$

- Convert letters to numbers: H $= 7$, E $= 4$, L $= 11$, O $= 14$.
- Apply $E(x)$:
  $E(H) = 10(K), E(E) = 7(H), E(L) = 14(O), E(O) = 17(R)$
- Encrypted message: **"KHOOR"**

Applications:

- Educational cryptography.
- Historical encryption (e.g., Julius Caesar).

Limitations:

- Easily broken using brute force or frequency analysis.

Substitution Cipher

Definition: A cryptographic technique that replaces each letter of plaintext with another letter based on a fixed mapping.

Encryption:

- Establish a one-to-one mapping between plaintext and ciphertext alphabets.
- Example Mapping: A Q, B W, ..., Z M.
- Plaintext: **SIMPLE** Ciphertext: **ZXQVWU**.

Decryption:

- Reverse the mapping to recover plaintext from ciphertext.

Applications:

- Military & Diplomatic Communication (Historical Use)
- Teaching Cryptography & Digital Security Concepts

Limitations:

- Vulnerable to Frequency Analysis
- Security Depends on Key Secrecy

Thank You