

# Developing an Efficient Bioinformatics Tool for Sequence Alignment

Nauman Sayed, Computer Science  
Mentor: David Claveau, Associate Teaching Professor  
School of Computing and Augmented Intelligence



## Background & Motivation

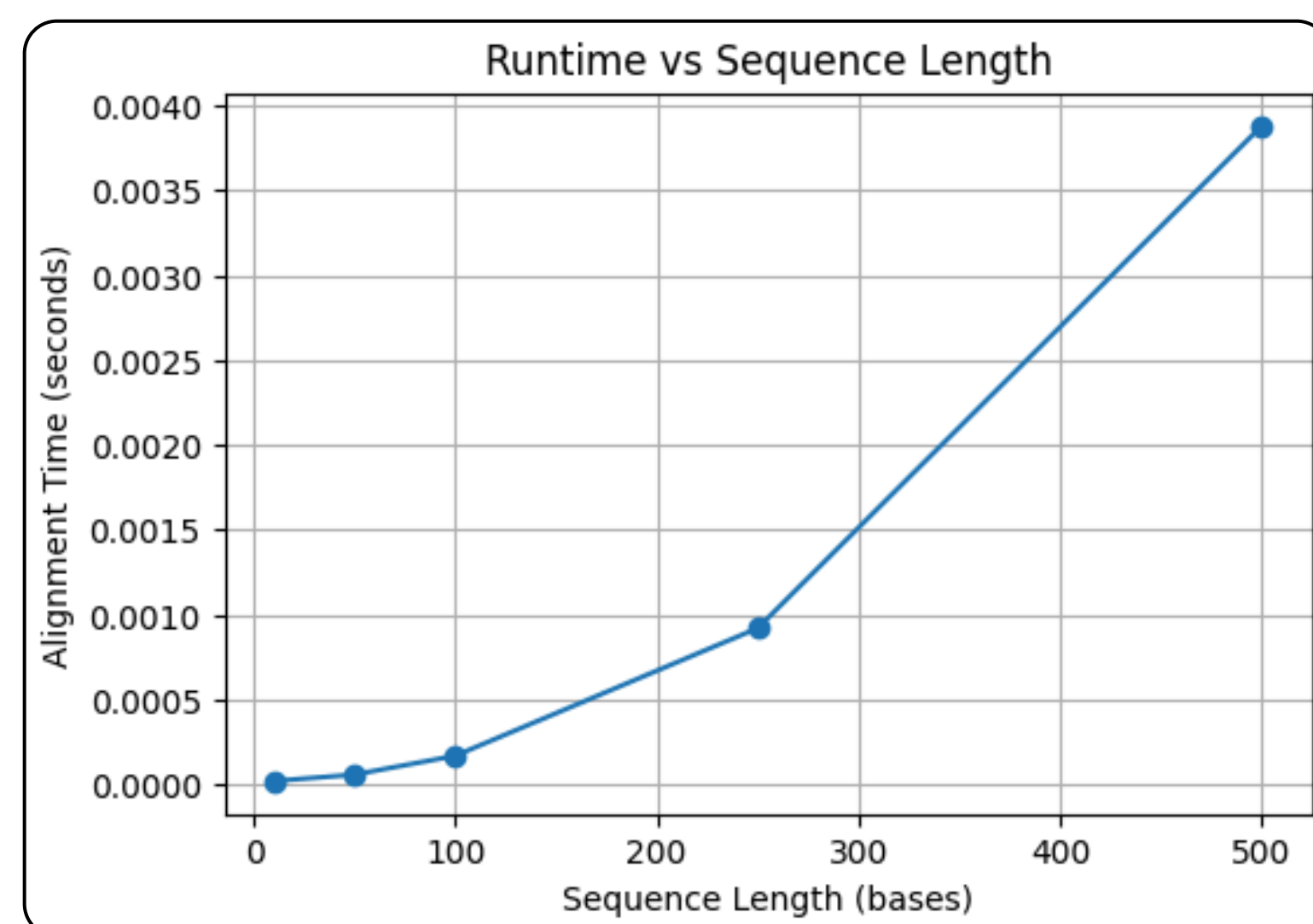
DNA alignment helps scientists find **genetic links**, **track mutations**, and **study diseases**.

Most existing tools are accurate but **slow for large DNA data**.

This project uses **Python** and **faster computing methods** to make DNA comparison quicker and more useful for **medical research**.

## Approach & Methodology

- Used **Biopython's PairwiseAligner** to align DNA sequences.
- Generated **random sequences** of different lengths.
- Tested **global** and **local alignment modes**.
- Measured **runtime** for **sequential** and **parallel** execution to compare efficiency.



Alignment time grows **quadratically** with sequence length ( $O(n^2)$ ).

## Results & Observations

*This code runs DNA sequence comparisons on **four CPU cores** at once to speed up analysis.*

```
from multiprocessing import Pool
from Bio.Align import PairwiseAligner
import random, time

aligner = PairwiseAligner()

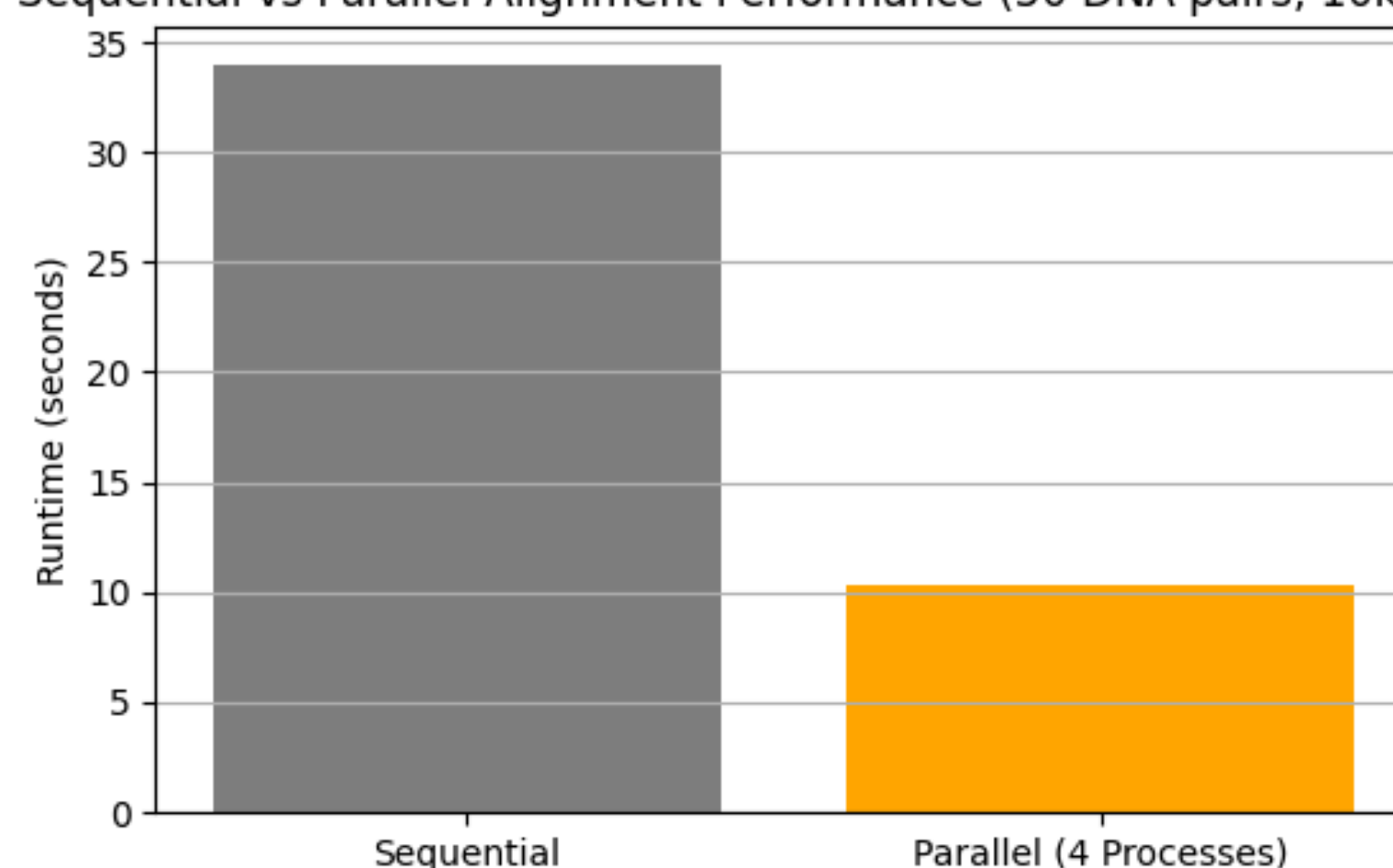
def rand_dna(n):
    return ''.join(random.choice('ACGT')
                    for _ in range(n))

def align_pair(pair):
    s1, s2 = pair
    return aligner.align(s1, s2)[0].score

pairs = [(rand_dna(10000), rand_dna(10000))
          for _ in range(50)]

with Pool(4) as p:
    start = time.time()
    results = p.map(align_pair, pairs)
    print("Parallel runtime:",
          round(time.time() - start, 2), "s")
```

Sequential vs Parallel Alignment Performance (50 DNA pairs, 10k bases)



*Multiprocessing achieved **~70% faster execution** for large datasets.*

## Discussion & Interpretation

- Sequence alignment scales with **quadratic time**, confirming expected algorithmic complexity.
- **Parallel processing** significantly reduced runtime, improving scalability for **large genomic datasets**.
- These findings support scaling genome analysis tools for **cloud** and **distributed computing**.

## Future Directions

- Implement **GPU-based parallelization** for faster computation.
- Add **BLOSUM/PAM scoring** for protein alignment.
- Test with **real genomic datasets** to validate performance.
- Develop a **graphical interface** for user accessibility.

## Acknowledgment

I would like to thank **Prof. David Claveau** for his guidance and mentorship on this project.