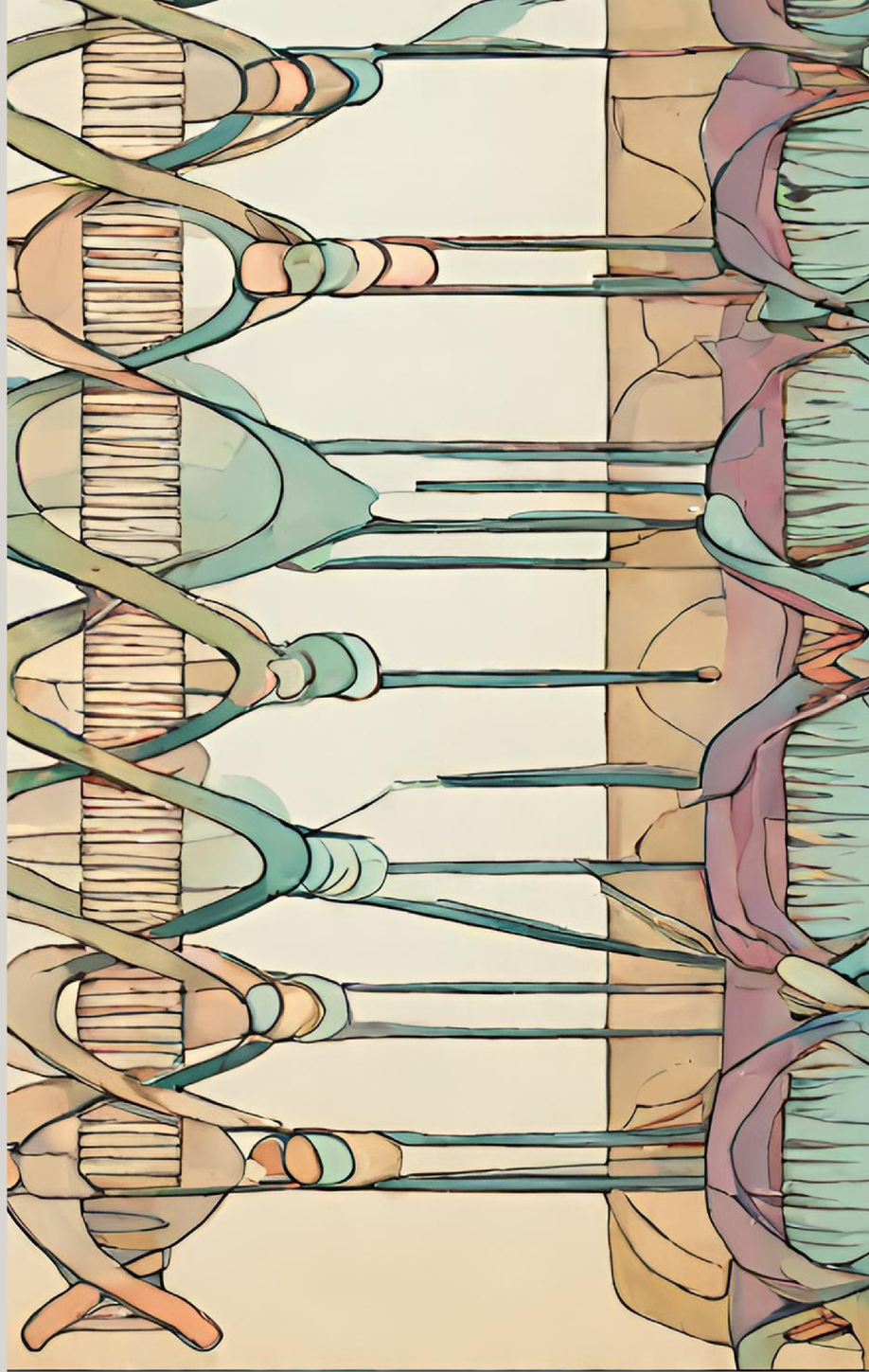
Design Credit Project -:

Topic-: Bioinformatics Tool for Sequence Alignment and De Bruijn Graph Analysis

Group members-:

Yashraj Chaturvedi(B22AI059)

Anuj Chincholikar(B22ES018)



CONTENTS

Introduction

- **Brief overview of the problem statement-:**
  - Sequence alignment using the Needleman-Wunsch algorithm and De Bruijn graphs construction for sequence assembly as well as analyzing genetic sequences along with substring computations for large dataset.

- **Objective of the Project:**
  - Substring Computation: Extracts substrings.
  - De Bruijn Graph Construction,Visualizations, Paths
  - Needleman-Wunsch Algorithm: Alignment using dynamic programming.
  - Alignment Integration with De Bruijn Graph: Efficiently identifies overlaps.
  - Similarity Percentage Calculation: Assesses alignment quality.

Importance of problem statement and Challenges in Field

Integration of De Bruijn Graph and Sequence Alignment:

- For comparing genetic sequences
- Prediction of functional regions in genomic DNA and Nucleotides
- Identification of similarities and differences among bps
- Aids in Evolutionary ,phylogenenic and mutation studies
- Efficient represention sequence data
- Facilitate genome assembly from short reads

Challenges:

- Coping with diverse genetic sequences and variations.
- Managing high computational demands for processing large datasets.
- Uncertainties and errors in alignments due to sequencing.
- Handling complexities such as repetitive regions and gene duplications in genomes.
- Time-consuming DNA alignment on real genomic datasets.

Various Algorithms for Alignment and Assembly

Genome Assembly:

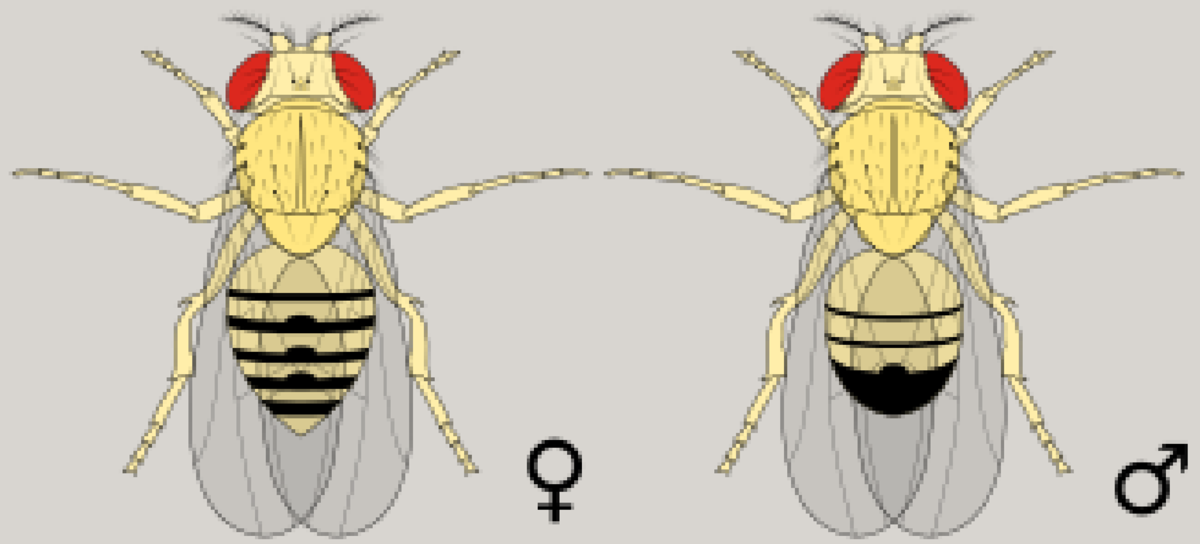
- **Naive Algorithm:**
  - A basic approach concatenating overlapping reads without considering errors or overlaps, leading to inaccurate results, especially in repetitive regions.
- **Greedy Algorithm:**
  - Assembling reads by extending contigs based on immediate overlaps, efficient but may lead to suboptimal assemblies, particularly in complex genomic regions.
- **De-bruijn Graphs**

Genome Alignment:

- **Smith-Waterman:**
  - Local alignment via dynamic programming, finding similar regions.
- **Needleman-Wunsch:**
  - Global alignment via dynamic programming, aligning entire sequences.
- **BLAST**
  - Heuristic local alignment tool for comparing sequences.
- **Hidden Markov Models (HMMs)**
  - Statistical models for aligning sequences with structural similarities.

Input File of Drosophila Melanogaster DNA

```
>Unmapped_Scaffold_8_D1580_D1567 dna:primary_assembly
primary_assembly:BDGP6.46:Unmapped_Scaffold_8_D1580_D1567:1:88768:1 REF
TAGAATAGAATAGAATAGAATAGAATATAATAGAATAGAATAGAATAGAATAGAATAGAA
TAGAATAGAATAGAATATAATAGAATATAATAGAATAGAATAGAATAGAATAGAATTGAA
AAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAA
TAGAATAGACTAGAATAGAATAGGATAGAATATAATAGAATAGAATAGAGAGAATAGAAT
AGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAAT
AGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAAT
AGAATAGAATAGAATAGAATAGAAATAGAATAGAATAGAATAGAATAGAATAGAATATAA
TAGAATAGAATAGAATGGAATAGAATATAATAGAATAGAACAGAATAGAATAGAATAGAA
TAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGACTAGAATAGAATAGAATAGAA
TAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAGTAGAATCGTA
TAGAGTAGAACAGGCTAGGATAGAATAGGACTAGAATGGAGTGGAATGGCATAGACTAGA
GTAGAGTAGAATAGAATAGGAATAGAATAGAATAGAATAGAGCAGAATCGAATAGAATAG
AATAGAATAGAATAGAATAGAATAGAATAGAATAGCAATAGAATAGAATAGAATAGAATA
GAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATA
GAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATA
GAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATA
GAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATA
GAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATA
GAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATA
GAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATAGAATA
```



ref. https://ftp.ensembl.org/pub/release-111/fasta/drosophila_melanogaster/dna/

- **Insects (Drosophila melanogaster):** Approximately 180 million base pairs (Mb), equivalent to 180,000,000 bp.
- **Humans (Homo sapiens):** Approximately 3,200 million base pairs (Mb), equivalent to 3,200,000,000 bp.

- **Assembly Types:** Primary, Nonchromosomal, Toplevel.
- **Sequence Characteristics:** dna, dna_rm, dna_sm.
- **Variability in Size and Timestamps:** Reflects data content, annotation levels, and currency.

https://ftp.ensembl.org/pub/release-111/fasta/drosophila_melanogaster/dna/

Index of /pub/release-111/fasta/drosophila_melanogaster/dna

Name	Last modified	Size	Description
Parent Directory		-	
CHECKSUMS	2023-10-19 16:11	2.3K	
Drosophila_melanogaster.BDGP6.46.dna.nonchromosomal.fa.gz	2023-10-04 12:42	1.5M	
Drosophila_melanogaster.BDGP6.46.dna.primary_assembly.2L.fa.gz	2023-10-04 12:42	6.9M	
Drosophila_melanogaster.BDGP6.46.dna.primary_assembly.2R.fa.gz	2023-10-04 12:42	7.3M	
Drosophila_melanogaster.BDGP6.46.dna.primary_assembly.3L.fa.gz	2023-10-04 12:42	8.2M	
Drosophila_melanogaster.BDGP6.46.dna.primary_assembly.3R.fa.gz	2023-10-04 12:42	9.3M	
Drosophila_melanogaster.BDGP6.46.dna.primary_assembly.4.fa.gz	2023-10-04 12:42	395K	
Drosophila_melanogaster.BDGP6.46.dna.primary_assembly.X.fa.gz	2023-10-04 12:42	6.8M	
Drosophila_melanogaster.BDGP6.46.dna.primary_assembly.Y.fa.gz	2023-10-04 12:42	889K	

Substring computation of large strings on Drosophila melanogaster DNA

- **Approach Used:**
  - Read genetic sequences from a FASTA file.
  - Iterate through the DNA sequence in steps equal to the desired substring length
  - Write computed substrings to a CSV file for further analysis.
  - Later this will be utilized in de-bruijn graph contruction

Output of computed substring in csv file

```
Enter the genetic sequence: ATGCTAGCTGATGCTGATGCTAGCGTAGCGTACTATCATCTCTACGTGCGAGGACTGACTGCTGATGCTGATCGAGCGAGCGCGCGCGCGCTCTTCTCTCGCGCGCGC
Enter the substring length: 5
Substrings of length 5 in the genetic sequence:
ATGCT
TGCTA
GCTAG
CTAGC
TAGCT
AGCTG
GCTGA
CTGAT
TGATG
GATGC
```

Output for small protein sequence

1 to 10 of 19510 entries Filter

Substring
AATGAATT
ATGAATTG
TGAATTGC
GAATTGCC
AATTGCCT
ATTGCCTG
TTGCCTGA
TGCCTGAT
GCCTGATA
CCTGATAA

Show 10 per page

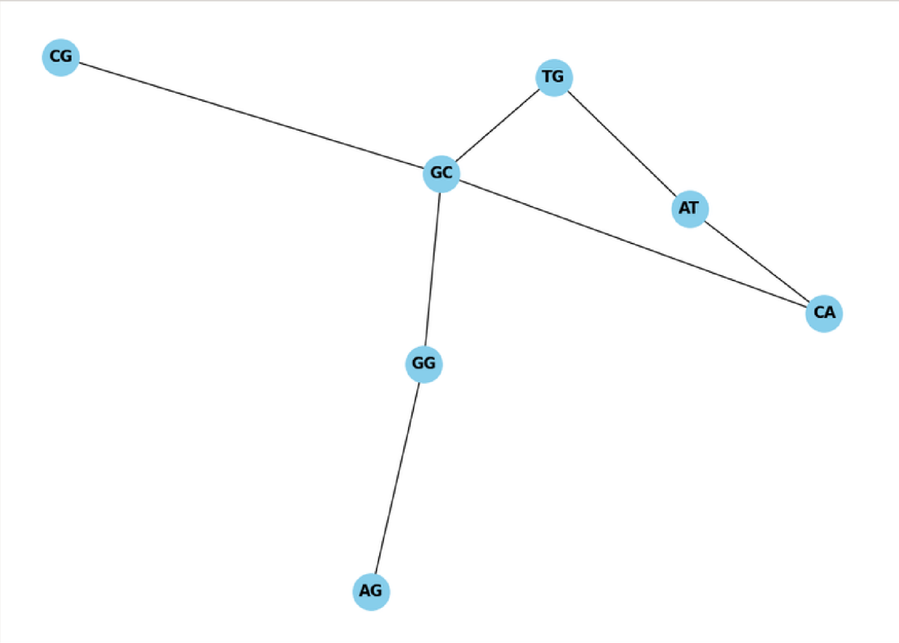
1 2 10 100 1000 1900 1950 19

There are a total of 19510 substring computed of k-mer length 8 for drosophila

Definitions and Importance: Overlapping , De-bruijn Graphs , Paths

- Overlapping graphs show how short DNA reads overlap, forming a network.
- De Bruijn graphs (a special case of overlapping graph) represent overlaps between substrings of fixed length in sequences.
    - They aid in genome assembly and reconstructing DNA sequences from short reads.
- Eulerian paths traverse each edge of a graph exactly once
    - They are used in genome assembly algorithms like Eulerian assembly as well as sequence analysis
- Hamiltonian paths visit each vertex of a graph exactly once.
    - They are applied in various optimizations and finding sub-sequence of particular length but less common in genomics due to complexity.

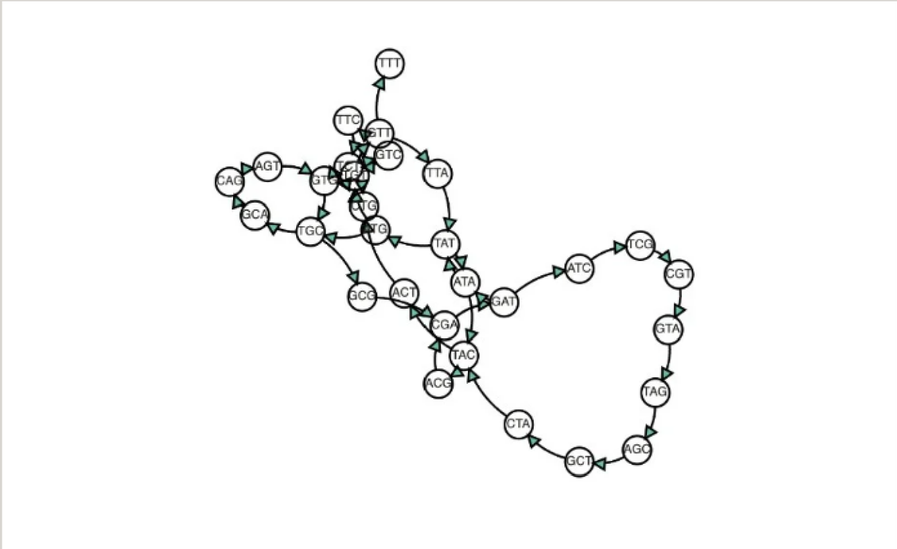
Visulaizations of De-brujin graphs obtained from small Reads




```
# Example reads
reads = ["ATGCG", "GCATG", "CATGC", "AGGCA", "GGCAT"]

# Length of k-mer
k = 3
```

dna_alignment_and_parallel_processing.py

```
ATCGATCGATCGATCGAATGTTCGATCGATCGATTTACTG
```



Output of Obtained Debrujin Edges,nodes, Eulerian path and hamiltonian paths for protein sequence.fasta file-:

sequence.fasta

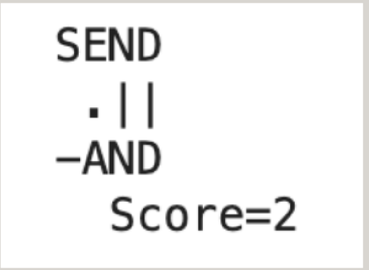
```
>AAH22532.1 Prion protein [Homo sapiens]
MANLGCWMLVLFVATWSDLGLCKKRPKPGGWNTGGSRYPGQGSPGGNRYPPQGGGGWGQPHGGGWGQPHG
GGWGQPHGGGWGQPHGGGWGQGGGTHSQWNKPSKPKTNMKHMAGAAAAGAVVGGLGGYVLGSAMSRPIIH
FGSDYEDRYYRENMHRYPNQVYYRPMDEYSNQNNFVHDCVNITIKQHTVTTTTKGENFTETDVKMMERVV
EQMCITQYERESQAYYKRGSSMVLFSSPPVILLISFLIFLIVG
```

```
De Bruijn Edges:
[('L', 'F'), ('C', 'V'), ('S', 'S'), ('D', 'R'), ('G', 'L'), ('I', 'H'), ('T', 'G'), ('H', 'R'), ('F', 'T'), ('N', 'K'),

De Bruijn Nodes:
['C', 'R', 'Y', 'G', 'T', 'V', 'W', 'I', 'L', 'F', 'A', 'N', 'Q', 'E', 'H', 'M', 'S', 'P', 'K', 'D']

Eulerian Path:
['C', 'W', 'M', 'A', 'N', 'L', 'G', 'C', 'K', 'K', 'R', 'P', 'K', 'P', 'G', 'L', 'V', 'L', 'F', 'V', 'A', 'T', 'W', 'S',

Overlapping Graph:
( L , F ) ( L , C ) ( L , V ) ( L , G ) ( L , L ) ( L , I ) ( C , V ) ( C , I ) ( C , K ) ( C , W ) ( S , S ) ( S , F ) (

Hamiltonian Path:
['C', 'V', 'G', 'L', 'F', 'T', 'K', 'M', 'S', 'A', 'N', 'R', 'Y', 'P', 'I', 'H', 'D', 'E', 'Q', 'W']
```

DNA ALIGNMENT

Needleman Wunsch Algorithm for global alignments on protein sequences

- **Needleman-Wunsch algorithm**: The Needleman_wunsch takes two input sequences seq1 and seq2, along with optional parameters for match, mismatch, and gap scores.
- **Initialization:** Creates score and traceback matrices.
- **Scoring Scheme**: Assigns scores for matches, mismatches, and gaps.We have taken match_score=1, mismatch_score=-1, gap_score=-2
- **Filling the Matrix**: Calculates scores for each cell using dynamic programming.
- **Traceback:** Reconstructs optimal alignment by following traceback path.
- **Alignment and Scoring**: Outputs aligned sequences and calculates alignment score.
- **Applications:** Used in DNA alignment to identify evolutionary relationships, functional elements, and mutation effects.
- **Implementation:** Reads sequences from a file, aligns pairs of sequences, calculates similarity percentage, constructs De Bruijn graphs, finds common paths, and visualizes graphs.
- **Importance:** Fundamental tool in bioinformatics for sequence analysis and similarity detection.



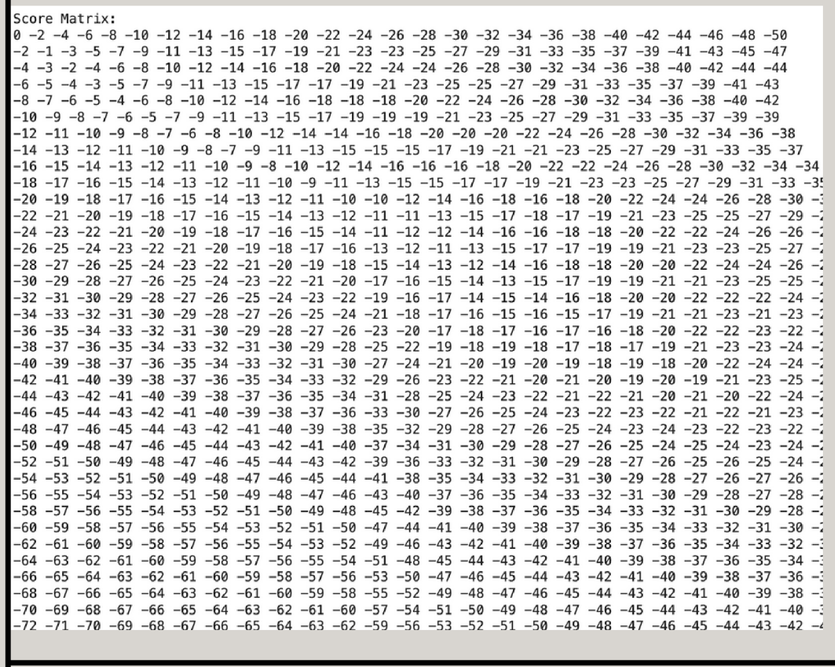
Input File of 11 Protein Sequences of different organisms

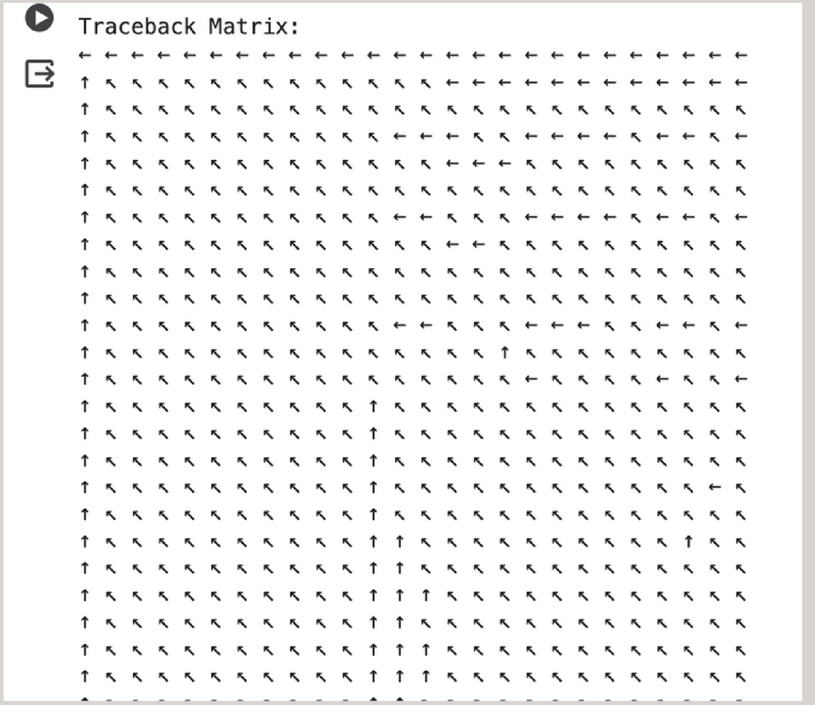
ref. https://www.ncbi.nlm.nih.gov/protein/

```
>Seq1 [organism=Carpodacus mexicanus] [clone=6b] actin (act) mRNA, par
CTTTATCTAATCTTTGGAGCATGAGCTGGCATAGTTGGAACCGCCCTCAGCCTCCTCATCCGTGCAGA
TTGGACAACCTGGAACTCTTCTAGGAGACGACCAAATTTACAATGTAATCGTCACTGCCCACGCCTTC
AATAATTTTCTTTATAGTAATACCAATCATGATCGGTGGTTTCGGAAACTGACTAGTCCCACTCATAA
GGCGCCCCCGACATAGCATTCCCCCGTATAAACAACATAAGCTTCTGACTACTTCCCCCATCATTTCT
TACTTCTAGCATCCTCCACAGTAGAAGCTGGAGCAGGAACAGGGTGAACAGTATATCCCCCTCTCGCT
TAACCTAGCCCATGCCGGTGCTTCAGTAGACCTAGCCATCTTCTCCCTCCACTTAGCAGGTGTTTCCT
ATCCTAGGTGCTATTAACTTTATTACAACCGCCATCAACATAAAACCCCCAACCCTCTCCCAATACCA
CCCCCCTATTCGTATGATCAGTCCTTATTACCGCCGTCCTTCTCCTACTCTCTCTCCCAGTCCTCGCT
TGGCATTACTATACTACTAACAGACCGAAACCTAAACACTACGTTCTTTGACCCAGCTGGAGGAGGAG
CCCAGTCCTGTACCAACACCTCTTCTGATTCTTCGGCCATCCAGAAGTCTATATCCTCATTTTAC

>Seq2 [organism=uncultured bacillus sp.] [isolate=A2] corticotropin (
GTAGGTACCGCCCTAAGNCTCCTAATCCGAGCAGAACTANGCCAACCCGGAGCCCTTCTGGGAGACGA
AAATCTACAACGTAGTCGTTACGGCCCACGCCTTCGTAATAATCTTTTTCATAGTAATGCCAATCATA
CGGAGGATTCGGGAACTGACTAGTTCCTCTAATGATTGGGGCCCCAGACATAGCATTCCCTCGAATAA
AACATAAGCTTTTGACTACTACCACCATCATTCCTACTCCTAATAGCCTCCTCAACAGTAGAAGCAGG
CCGGAACCGGATGAACCGTGTACCCACCACTAGCTGGAAACCTGGCCCACGCCGGAGCCTCAGTAGAC
AGCTATCTTCTCCCTACACCTAGCAGGTATCTCATCCATCCTGGGGGCAATTAACTTCATTACAACAG
ATCAACATAAAACCACCCGCCCTCTCACAATACCAAACACCACTATTCGTGTGATCCGTCCTAATTAC
CCGTACTACTCCTACTATCTCTCCCAGTACTAGCCGCCGGTATCACCATGCTACTCACAGACCGCAAC
CAACACCACCTTCTTTGACCCAGCAGGAGGAGGAGACCCAGTACTATACCAGCACCTATTCTGATTCT
GGACACCCAGAAGTCTACATCCTAATTCTC

>Seq3 [organism=Phalaenopsis equestris var. leucaspis]
CTATACCTAATTTTCGGCGCATGAGCCGGAATGGTGGGTACCGCTCTAAGCCTCCTCATTCGAGCAGAA
TAGGCCAACCCGGAGCCCTTCTGGGAGACGACCAAGTCTACAACGTGGTTGTCACGGCCCATGCCTTC
AATAATCTTCTTTATAGTTATGCCGATTATAATCGGAGGATTCGGAAACTGACTAGTCCCCCTAATAA
GGAGCCCCAGACATAGCATTTCCGCGAATAAACAACATAAGCTTCTGACTACTCCCACCATCATTCCT
TCCTCTTAGCATCCTCCACAGTGGAAGCAGGCGTAGGTACAGGCTGAACAGTGTATCCCCCACTAGCT
CAACCTAGCTCATGCCGGGGCCTCAGTCGACCTCGCAATCTTCTCCTTACACCTAGCTGGTATTTCCT
ATCCTCGGAGCAATTAACTTCATTACAACAGCAATTAACATGAAACCTCCTGCCCTCTCACAATACCA
CCCCACTATTCGTCTGATCAGTGTTAATTACTGCAGTCCTCCTTCTCCTTTCCCTTCCAGTTCTAGCT
AGGAATCACAATGCTCCTCACAGACCGCAACCTCAACACCACATTCTTCGACCCTGCCGGAGGAGGAG
CCCGTCCTATATCAACATCTCTTCTGATTCTTCGGCCACCCAGAAGTCTACATCCTAATCCTC

>Seq4 [organism=uncultured archaeon]
ATGAGCTGGAATAGTAGGTACCGCCCTAAGCCTCCTAATTCGAGCAGAGCTAGGCCAACCCGGAGCCC
CTGGGAGACGACCAAATCTACAACGTAGTCGNCACGGCCCATGCTTTTGTAATAATCTTCTTCATAGT
TGCCAATCATAATCGGAGGGTTTGGAAACTGACTGGTCCCCCTAATAATTGGAGCTCCAGACATAGCA
```

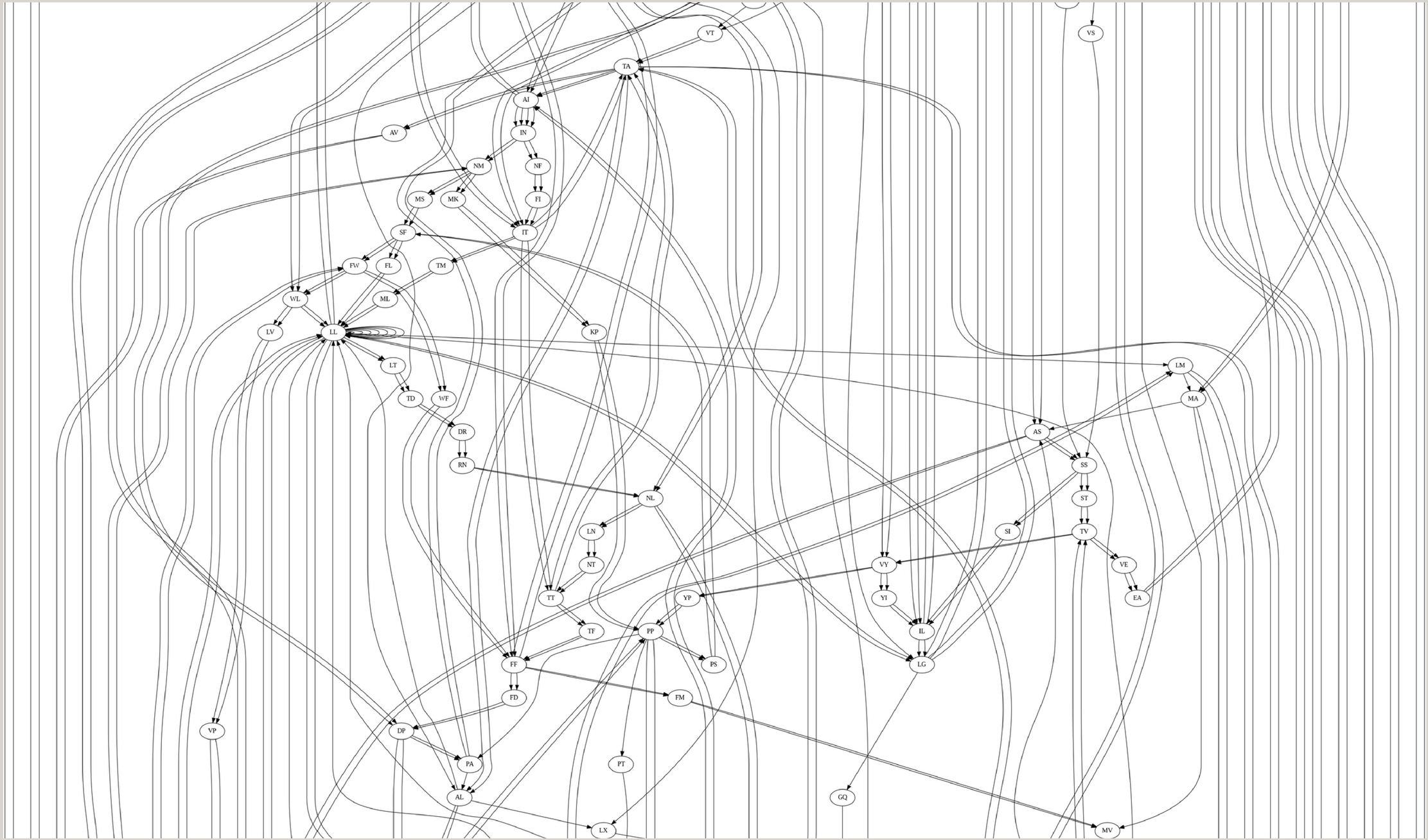
Score and Traceback matrices

- **Score Matrix:**
  - Represents scores of aligning subsequences.
  - Initialized with scores based on a scoring scheme.
  - Filled iteratively based on adjacent cell scores and character similarities.
- **Trace-back Matrix:**
  - Guides backtracking from bottom-right to top-left.
  - Each cell indicates direction: diagonal (match/mismatch), up (gap in first sequence), or left (gap in second sequence).
  - Used to reconstruct the optimal alignment.





Visulatization of Debruijn graphs on protein dataset using graphviz library



Output genreated from Needleman Wunsch algorithm when we applied it on Protein.txt file

```
Alignment and De Bruijn graph for sequences 1 and 2:
Alignment for sequences 1 and 2:
LYLIFGAWAGMVGTALSLLIRAELGQPGTLLGDDQIYNVIVTAHAFVMIFFMVMPIMIGGFGNWLVPLMIGAPDMAFPRMNNMSFWLLPPSFLLLLASSTVEAGAGTGWTVYPPLAGNLAH
           |||||.|||||||.|||.||||||||||.|||||||||||||||||||||||||||||||||||||||||||||||||||.|||||||||||||||||||||||
----------VGTALXLLIRAELXQPGALLGDDQIYNVVVTAHAFVMIFFMVMPIMIGGFGNWLVPLMIGAPDMAFPRMNNMSFWLLPPSFLLLMASSTVEAGAGTGWTVYPPLAGNLAH
  Score=184

similarity percentage from our model: 92.20779220779221
Similarity Percentage from library: 92.20779220779221
Common Paths: [('VGT', 'GTA'), ('GTA', 'TAL'), ('LSL', 'SLP'), ('LLI', 'LIR'), ('LIR', 'IRA'), ('IRA', 'RAE'), ('RAE', 'A
```

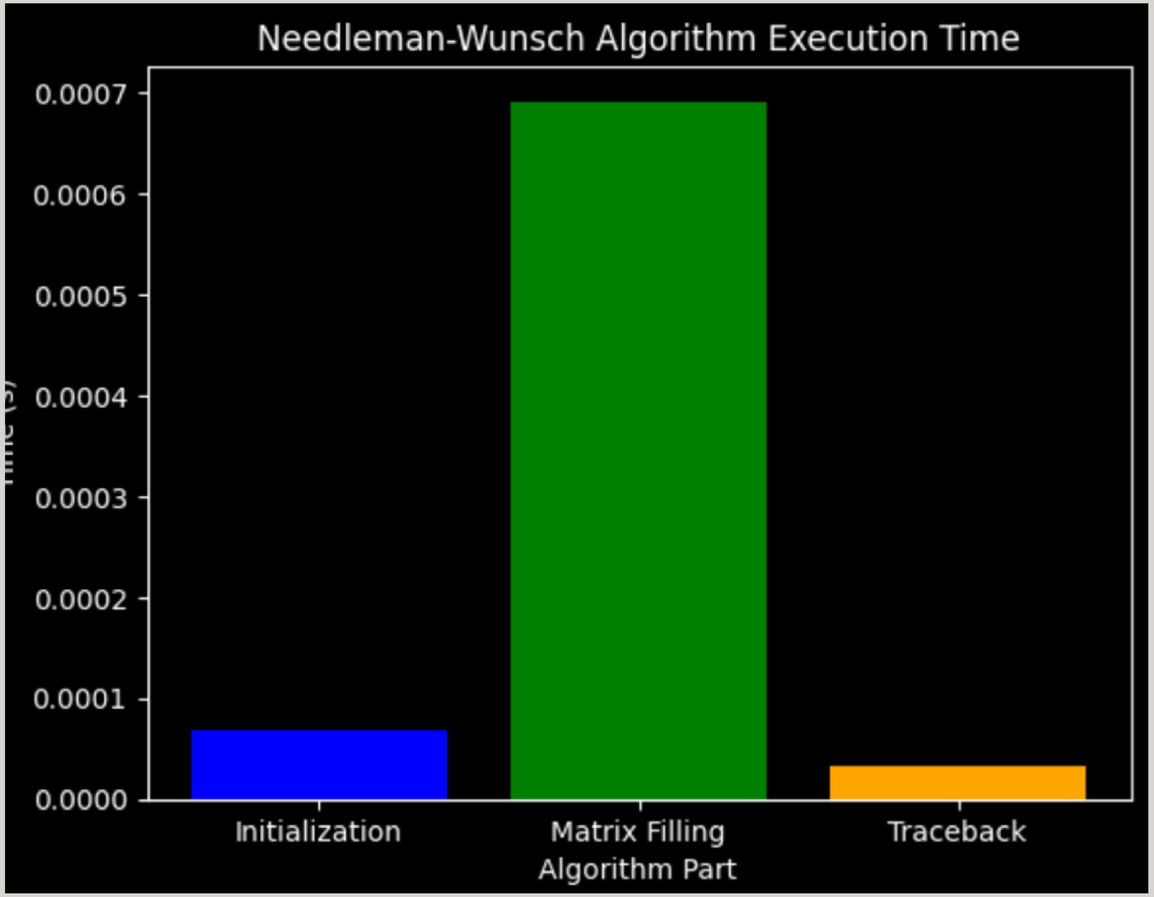
This is the image for alignment ,similarity percentage and common paths between sequence 1 and sequence 2 similarly there are total of 7 sequences so 7c2 pairs of comparisions.

```
De Bruijn graph:
LY -> YL, YQ, YQ
YL -> LI
LI -> IF, IR, IT, IL, IR, IT, IL
IF -> FG, FF, FS, FF, FS
FG -> GA, GN, GH, GN, GH
GA -> AW, AP, AG, AS, AI, AL, AP, AG, AS, AI
AW -> WA
WA -> AG
AG -> GM, GA, GT, GN, GA, GV, GI, GG, GA, GT, GN, GA, GI, GI, GG
GM -> MV
MV -> VG, VM, VM
```

We also computed cumulative error percentage in our model with respect to library function “pairwise2” in Bio , which came out to be 0.04409549123602203% in this case.

```
% error in our model 0.04409549123602203%
```

Time taken by each step of Needleman wunsch algorithm on protein data



It is clearly from graph that Matrix filling step in needleman wunsch algorithm takes the maximum time to compute relative to initialization and traceback.

We can optimize this by parallel processing.

Conclusion

In this project we started off with base idea of how to compute substrings on large genetic sequences ,then we moved over to concept of de-brujin graphs , later on which we connected to overlapping graphs , we calculated eulerian and hamiltonian paths for these graphs and then started with concept of DNA Alignment wehere we did Needlenman Wunsch Algorithm which is a global alignment technique ,we saw similarity percentage between any 2 DNA sequences , made de-Brujin graph of those sequences to find common paths between both and lastly we saw time analysis of our algorithm on a Protein real time Dataset

Thank you!