Sequence Alignment Methods in Bioinformatics for Biological Sequences

(Needleman-Wunsch Method)

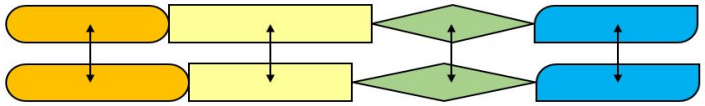
Group 7: Pratik Mistry, Vikhyat Dhamija, Aditya Singh Thakur

Agenda

- Introduction
- Terminology
- Needleman-Wunsch Method
  - Methodology
  - Pseudocode
- Complexity
- Demo
- Experiments
- Analysis
- Applications
- Related Work
- Future Scope
- Conclusion

Introduction (1)

- Sequence alignment method in bioinformatics used for arranging the sequences of DNA, RNA, or protein
- Regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences
- Needleman-Wunsch algorithm is the application of dynamic programming to find the optimal global alignments between two biological sequences
- Aligns every residue in every sequence when the sequences in the query set are similar and of roughly equal size



Global Alignment

Introduction (2)

- General steps to find optimal global alignment:
  - Determine the substitution matrix and scoring schemes - Match, Mismatch and Indel/Gap scores
  - Constructing and initializing the scoring matrix
  - Fill in the scoring matrix with scores in each cell
  - Traceback to find the global alignments
  - Choosing optimal global alignment

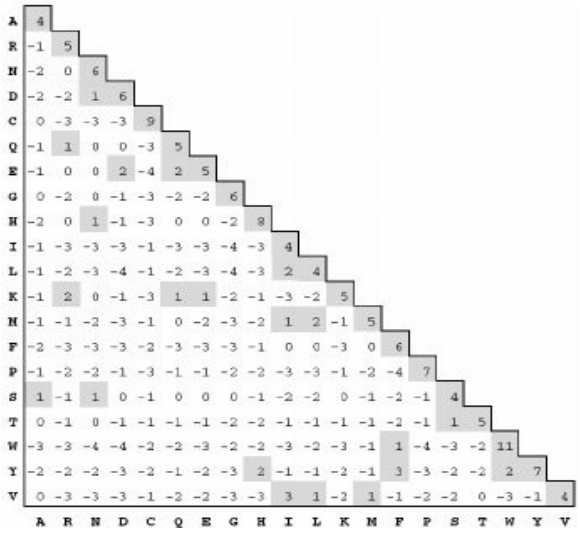
- Optimal global alignment is based on the highest alignment score calculated between two globally aligned sequences

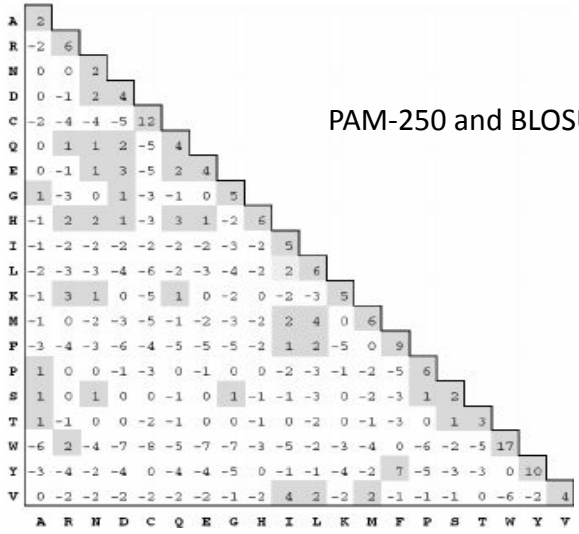
- Thus, when new sequence is found, the structure and function can be easily predicted by doing sequence alignment

Terminology (1)

- ***Biological Sequence***: A biological sequence is a single, continuous molecule of nucleic acid or protein. E.g. : G A A T T C

- ***Substitution Matrix:*** A grid that provides scores for the substitution of every amino acid (or nucleotide) for every other. Example: PAM and BLOSUM





PAM-250 and BLOSUM-62 matrices

Terminology (2)

- ***Scoring Schemes:*** A scoring scheme/system helps score each individual pair of letters. Example: Scores for Match between letters can be +1, Mismatch and Gap/Indel can be -1

- ***Scoring Matrix:*** Scoring Matrix is used to determine the relative score made by matching two characters in a sequence alignment

	A	C	G	T
A	1	-1	-1	-1
C	-1	1	-1	-1
G	-1	-1	1	-1
T	-1	-1	-1	1

- ***Gaps and Gap Penalty***: Gaps i.e. indels are gaps that may be created between proteins (letters) after finding optimal alignment. Gap penalty is the score that should be considered for each gap while calculating the score of an alignment.

Terminology (3)

- ***Traceback:*** Help find the alignment sequences between two sequences from a scoring matrix. Starts with the rightmost bottom cell of the scoring matrix and ends at the origin i.e. leftmost top cell

- ***Global Sequence Alignment:*** The best alignment over the entire length of two sequences when the two sequences are of similar length, with a significant degree of similarity throughout.

```
S I M I L A R I T Y
| | | | | | | | | |
P I - L L A R - - -
```

Needleman-Wunsch Method

- Applications of dynamic programming used to align and compare biological sequences
- Divides a large problem (e.g. the full sequence) into a series of smaller problems, and it uses the solutions to the smaller problems to find an optimal solution to the larger problem
- Assigns a score to every possible alignment, and hence finds all possible alignments having the highest score
- Good alignment has minimum gaps and mismatches inserted into the sequences, but at the same time maximize the number of positions in the aligned strings that match
- Example Sequences:
  Sequence A = SEND
  Sequence B = AND

Methodology (1)

Determining substitution matrix and Scoring Schemes: Critical step which makes a huge difference in results of alignment

Substitution Matrix

- PAM (Point Accepted Mutation) or BLOSUM (BLOcks SUbstitution Matrix) predefined matrices
- Helps us determine the score between pairs of protein elements.

Scoring Schemes

- Helps us in fill scoring matrix and also calculate scores of alignments
- Helps us traceback to help us choose best sequence alignment
- Example: Match = 1, Mismatch = -1, Gap or Indel = -1
- Typically, PAM and BLOSUM is used for determining **Match** value

Methodology (2)

Constructing and Initializing the scoring matrix: Helps in alignment process by storing scores

Constructing the Scoring Matrix

- Size is (m+1,n+1) where m and n are lengths of sequence strings A and B

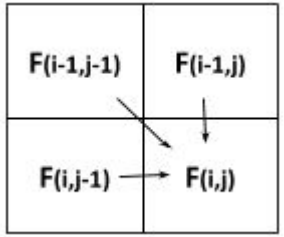
Initializing the Scoring Matrix

- First cell value = Origin = F(0,0) = 0
- Gap Penalty = -1
- Since, no 'top' or 'top-left' cells for the 1st row & only the existing cells to the left can be used to calculate the score of each cell in 1st row
- Thus, 1st row cells values = 0,-1,-2,-3,-4…
- Similarly, 1st column cells value = 0,-1,-2,...
- Initialization completely depends on the ***gap*** penalty or ***Indel*** value

		A	N	D
	F(0,0)	F(0,1)	F(0,2)	F(0,3)
S	F(1,0)	F(1,1)	F(1,2)	F(1,3)
E	F(2,0)	F(2,1)	F(2,2)	F(2,3)
N	F(3,0)	F(3,1)	F(3,2)	F(3,3)
D	F(4,0)	F(4,1)	F(4,2)	F(3,3)

		A	N	D
	0 →	-1 →	-2 →	-3
S	-1 ↓	F(1,1)	F(1,2)	F(1,3)
E	-2 ↓	F(2,1)	F(2,2)	F(2,3)
N	-3 ↓	F(3,1)	F(3,2)	F(3,3)
D	-4	F(4,1)	F(4,2)	F(4,3)

Methodology (3)



Fill in the scoring matrix

- Calculate scores of each cell of the matrix starting with cell F(1,1)
- Score of any cell F(i,j) is represented by formula

$$F(i,j) = \max \begin{cases} F(i-1,j-1) + S(A_i,B_j) \\ F(i-1,j) + g \\ F(i,j-1) + g \end{cases}$$

where $0 <= i <= m$ and $0 <= j <= n$

Where, g is the gap penalty,
F(i-1,j-1) is score of top left diagonal cell,
F(i-1,j) is score of upper cell
F(i,j-1) is score of left cell
S(Ai,Bj) is the match or mismatch score between elements at Ai and Bj

- S(Ai,Bj) is referred from substitution matrix like PAM or BLOSUM in real world
- For e.g: Since Match = 1, Mismatch = -1 and Gap = -1, F(1,1) is calculated as:

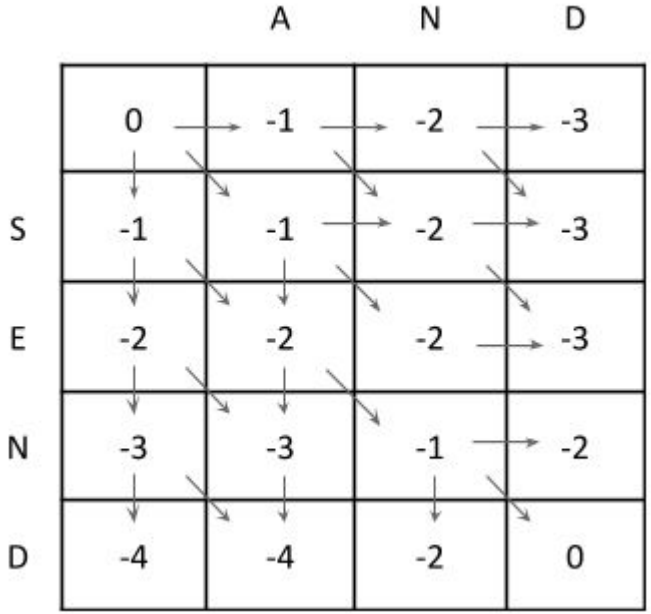
$$F(1,1) = \max \begin{cases} F(i-1,j-1) + S(A_i,B_j) = 0-1 = -1 \\ F(i-1,j) + g = -1-1 = -2 \\ F(i,j-1) + g = -1-1 = -2 \end{cases} = -1$$

S \ A		
	0	-1
	-1	**-1**

Methodology (4)

- Similarly, we calculate scores of all the cells of the scoring matrix

  NOTE: The arrows indicate from which cell (left, top or diagonal) the value of the current cell can be determined.



Methodology (5)

Traceback to find global alignments

- Find all the global alignments by traceback method where we start from the last cell of matrix and trace to origin i.e. from F(4,3) to F(0,0)

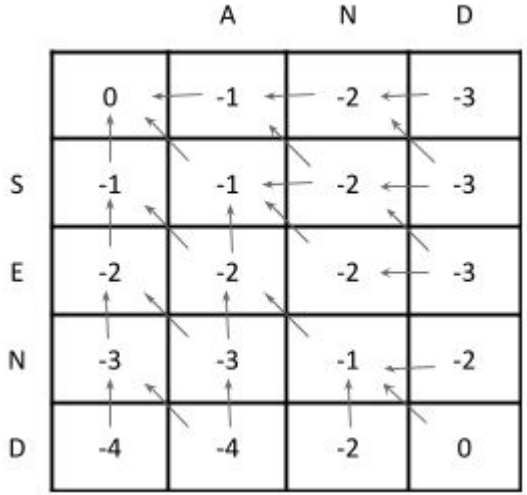
- Rules to consider while constructing sequences:
  - A diagonal arrow represents a match or mismatch, so the letter of the column and the letter of the row of the origin cell will align
  - A horizontal or vertical arrow represents an indel. Horizontal arrows will align a gap ("-") to the letter of the row (the "side" sequence), vertical arrows will align a gap to the letter of the column (the "top" sequence)
  - If there are multiple arrows to choose from, they represent a branching of the alignments. If two or more branches all belong to paths from the bottom right to the top left cell, they are equally viable alignments

Methodology (6)

- The scoring/traceback matrix can be represented as:
  NOTE: The arrows indicate from which cell (left, top or diagonal) the value of the current cell can be determined. Arrows are reversed as we perform traceback

- Based on the rules and scoring matrix the aligned sequences by exhaustive method of traceback are:

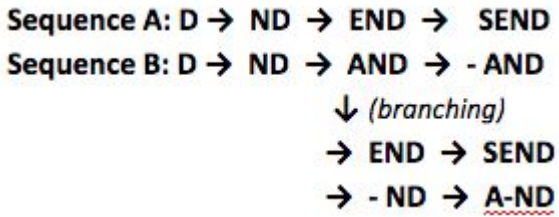
Sequence A: D → ND → END → SEND
Sequence B: D → ND → AND → - AND
↓ *(branching)*
→ END → SEND
→ - ND → A-ND



Methodology (7)

Choosing optimal global alignment

- Find the scores of each alignment and find optimal i.e. best global alignment based on scoring schemes



- For the first alignment:
  Score = Match * 2 + Mismatch * 1 + Gap * 1 = (1)(2) + (-1)(1) + (-1)(1) = **0**
- Similarly, for the second alignment: **Score = 0**
- Since the maximum score is 0, we can choose either of them as the optimal global alignment sequence
- The exhaustive method complexity in terms of compute and time increases with larger protein sequences as there would be many branching and resulting alignments
- Needleman-Wunsch algorithm uses dynamic programming technique and directly gives us the optimal global alignment in very less time instead of finding all the sequences

Pseudocode: Algorithm (1)

Algorithm 1: (Filling alignment matrix)

- Basis:
  F(i,0) = g * i
  F(0,j) = g * j
- Recursion, based on the principle of optimality:
  F(i,j) = max (F(i-1,j-1) + S(Ai,Bj), F(i-1,j) + g, F(i,j-1) + g)
- Initializing and filling the scoring matrix Fi,j where A and B are the protein sequences and S is the substitution matrix
- After scoring matrix F is computed, there would be an alignment sequence whose score would be maximum

Pseudo-code for the algorithm 1

```
g ← Gap Penalty          // Gap penalty

for i=0 to length(A):    // First column
        F(i,0) ← d*i

for j=0 to length(B):    // Second column
        F(0,j) ← d*j

for i=1 to length(A):    // Filling matrix
        for j=1 to length(B) {
                F(i,j) ← max(F(i−1,j−1) + S(Ai,
        Bj), F(i−1, j) + g,  F(i, j−1) + g)
 }
```

Pseudocode: Algorithm (2)

Algorithm 2: (Traceback for alignment seq.)

- Start the bottom right cell, and compare the value with the three possible cells (Top-left Diagonal, Left, and Top) to see which it is derived from as depicted with arrows
- Move till the origin i.e. top-left element 0
- Alignment scenarios:
  - If derived from Top Left Diagonal cell, then Ai and Bj are aligned,
  - If from the top cell, then Bj is aligned with a gap, and
  - If from the left cell, then Ai is aligned with a gap
- More than one choice leads to alternative optimal alignments

Pseudo-code for the algorithm 2

```
AlignmentA ← "",   AlignmentB ← ""
i ← length(A) ,    j ← length(B)
while (i > 0 or j > 0) {
 if (i > 0 and j > 0 and F(i,j) == F(i−1,j−1) + S(Ai, Bj)) {
        AlignmentA ← Ai + AlignmentA
        AlignmentB ← Bj + AlignmentB
        i ← i − 1
        j ← j − 1 }
 else if (i > 0 and F(i,j) == F(i−1,j) + d) {
        AlignmentA ← Ai + AlignmentA
        AlignmentB ← "−" + AlignmentB
        i ← i − 1 }
 else {
        AlignmentA ← "−" + AlignmentA
        AlignmentB ← Bj + AlignmentB
        j ← j − 1  }}
```

Complexity

Time Complexity: **O(mn)**
Space Complexity: **O(mn)**

Brute Force/Exhaustive Search Method v/s Needleman-Wunsch Algorithm:

- Complexity increases with larger protein sequences because of more branching and resulting alignments
- Number of possible global alignments between two sequences of length N

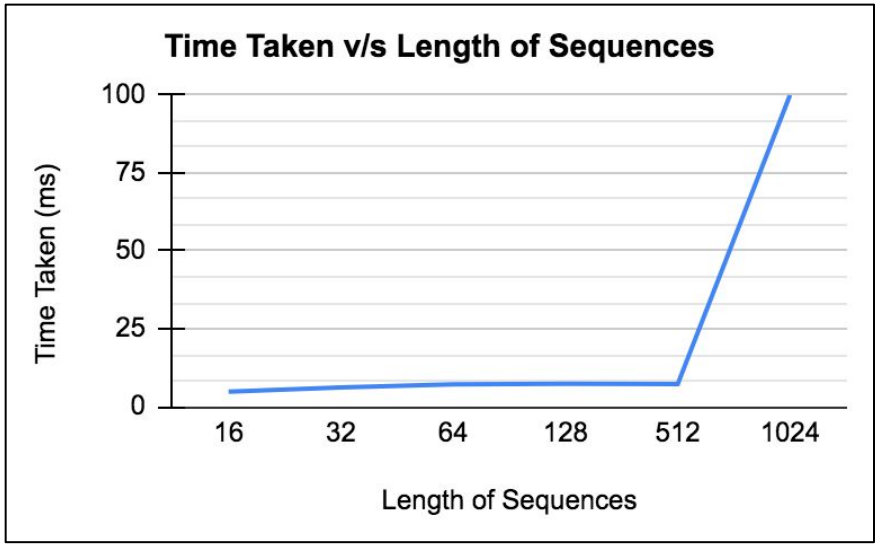
$$22N/\sqrt{\pi N}$$

- For eg. two sequences of 250 residues, it is ~ 10^149 which is a NP hard problem
- Needleman-Wunsch requires filling a 250 × 250 matrix and returns the optimal global alignment in lesser time complexity because of dynamic programming
- Needleman-Wunsch works regardless of the length or complexity of sequences, and guarantees to find the best alignment.

Demo

Experiments and Analysis

Length of Sequence	Time Taken (ms)
16	4.89
32	6.22
64	7.25
128	7.35
512	7.27
1024	Out of Memory



Applications

Applications of Needleman Wunsch Algorithm (NWA)

- **Uses in Computer Vision** - NWA can be used in the 3D Reconstruction process which uses Stereo matching, it can be done by matching pixels belonging to scan lines, since it aims at establishing optimal correspondence between two strings of characters.
- **Uses in Bioinformatics** - to align protein and nucleotide sequences
  The alignment sequences help us in study of genomics,
  compare biological sequences,
  find relationships and homology between different sequences,
  and can provide useful information, which can be used to further identify new members of protein families.
  Applications in identifying DNA mutations in Coronavirus
  Reference Link - https://iopscience.iop.org/article/10.1088/1742-6596/1218/1/012031/pdf
- **Uses in Process Mining**

Related Work

- **Smith-Waterman algorithm** is another sequence alignment method that performs *local sequence alignment* which is necessary for determining similar regions between two strings of nucleic acid sequences or protein sequences

- **Levenshtein distance algorithm** is a string metric algorithm for measuring the difference between two sequences

- It has also been shown that it is possible to improve the running time to O(mn/log n) using the **Method of Four Russians technique** - a variation in Needleman-Wunsch algorithm.

- **Process-oriented Iterative Multiple Alignment** for Medical Process Mining used mainly for workflow analysis, medical traces/process alignment and knowledge discovery.

Future Scope

- New Parallel approach of Needleman Wunsch Algorithm for global sequence alignment
- Comparing various different parallel approaches and their execution time in order to determine the fastest and the most efficient approach . For example - sequential CPU with parallel GPU
- Experimenting with different kernel calls, shared memory , anti - diagonal in parallel using threads etc so that we can improve the performance of the project.

Problems Faced

- Understanding and implementing the algorithm
- Difficulty in finding a proper machine with suitable specifications (GPU) to execute the CUDA program
- Problems faced while implementing Orbits

Conclusions

- In this project, we discussed the detailed steps with analysis of Needleman-Wunsch algorithm for biological sequence alignment used widely in the field of Bioinformatics for study of genes, protein sequences and DNA/RNA using CUDA
- The Needleman-Wunsch algorithm is appropriate for finding the best alignment of two sequences which are (i) of the similar length; (ii) similar across their entire lengths. This work has a very great impact in life sciences as it helps in finding the newer sequences or helps us find the homology relationships with the existing protein sequences by finding optimal global alignment sequences.
- We discussed the wide applications of algorithms in other business domains making it very popular.
- Demonstration of the working of the project.
- Discussed related work done in the similar field, and lastly
- Problems faced and Future Scope on how our project can be improved.

References

- https://en.wikipedia.org/wiki/Bioinformatics
- https://en.wikipedia.org/wiki/Needleman%E2%80%93Wunsch_algorithm
- https://www.cs.sjsu.edu/~aid/cs152/NeedlemanWunsch.pdf
- https://en.wikipedia.org/wiki/Substitution_matrix
- https://www.researchgate.net/figure/PAM-250-and-Blosum-62-matrices_fig1_265241022
- https://en.wikipedia.org/wiki/Point_accepted_mutation
- https://en.wikipedia.org/wiki/BLOSUM
- https://en.wikipedia.org/wiki/Smith%E2%80%93Waterman_algorithm#Algorithm
- https://en.wikipedia.org/wiki/Multiple_sequence_alignment
- https://en.wikipedia.org/wiki/Levenshtein_distance
- https://en.wikipedia.org/wiki/Method_of_Four_Russians
- “Process-oriented Iterative Multiple Alignment for Medical Process Mining”, S Chen, S Yang, M Zhou, R Burd, I Marsic, 2017 IEEE International Conference on Data Mining Workshops (ICDMW), 438-445
- https://en.wikipedia.org/wiki/Sequence_alignment

Thank you

Q&A