Align that Sequence!

Sonal Joshi 6540-9104, Aditya Kulkarni 8961-9483
sonal.joshi@ufl.edu, a.kulkarni@ufl.edu
Bioinformatics, Department of Computer and Information Science and Engineering,
University of Florida, Gainesville, Florida.

***Abstract*- Sequence Alignment is an arrangement of DNA, RNA or protein sequences to identify similar or nearly similar sequences. This not only helps in identifying the similar functionality of the sequences but also helps in understanding the structure and evolutionary relationships. Needleman-Wunsch and Smith-Waterman[4] are two popular algorithms used for dovetail sequence alignment. We are limiting our project to Needleman-Wunsch algorithm. It uses dynamic programming to find the maximum score a sequence could have. Using this sequence as a best case we allow users to try and match the score by dynamically aligning the sequence on the UI.**

Index Terms*: *Sequence Alignment, Needleman-Wunsch, Dynamic Programming, Scoring matrix

I. INTRODUCTION

Sequence alignment[2][1] is a way of arranging DNA, RNA or protein sequences to identify the similar regions in the long sequences. This alignment can help us understand the functionality, structure and evolutionary relationship between the two sequences. Aligned sequences of amino acids or nucleotides are arranged into matrices and gaps are inserted to align the sequences. Dynamic programming is extensively used in this process.

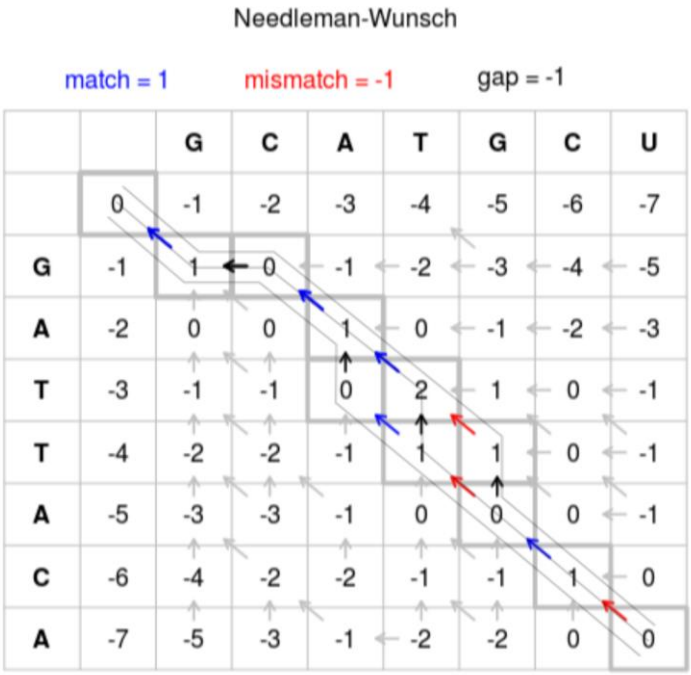
In this project we have implemented Needleman-Wunsch algorithm[4] which uses dynamic programming to calculate the scoring matrix. It is widely used for global alignment. The algorithm assigns score to every possible combination of the sequence. There is a positive score for a match and negative score for mismatches which also includes gaps.

Using this algorithm as a base for a web application we have developed an interactive platform where users can learn about the algorithm and understand the working of the algorithm through a game like setup.

II. DYNAMIC PROGRAMMING METHOD

To find the maximum possible score between two randomly generated sequences we need to create a matrix of dimensions MxN, where M and N are the lengths of the two sequences respectively.

Let us look at an example of the algorithm.



In this example we have the match score as 1, mismatch score as -1 and a gap score is also -1. These scores can be changed real time based on the nature of study[4].

First row and first column is used for the initialization of the algorithm. The pseudo code for the algorithm is as follows.

```
d ← MismatchScore
for i=0 to length(A)
  F(i,0) ← d*i
for j=0 to length(B)
  F(0,j) ← d*j
for i=1 to length(A)
  for j=1 to length(B)
  {
    Match ← F(i-1,j-1) + S(Ai, Bj)
    Delete ← F(i-1, j) + d
    Insert ← F(i, j-1) + d
    F(i,j) ← max(Match, Insert, Delete)
  }
```

Algorithm -

Since each match and mismatch can be assigned a separate score, this information is used to assign values in the first row and column. As we further traverse each row to check if the character in the row matches with the character in the column, we try to associate a score to it using the match/mismatch score and gap score values. To elaborate on the same, assume if there is a match, then we add the matched score to the previous diagonal value and check with the other two values obtained by deleting this element i.e we add the gap score to the other two. Once all three values have been calculated we take the maximum of the three values and assign it to the current position in matrix[4]. Matrix is traversed till the end and the maximum value is obtained. This maximum value is the maximum matching score which this sequence could have.

In this project, we deal with two randomly generated sequences. The second sequence is generated by mutating the first sequence with a pre-defined mutation probability. Mutation can either be a replacement or deletion of letter(A,C,G,T).

Code snippet for Mutation of sequences.

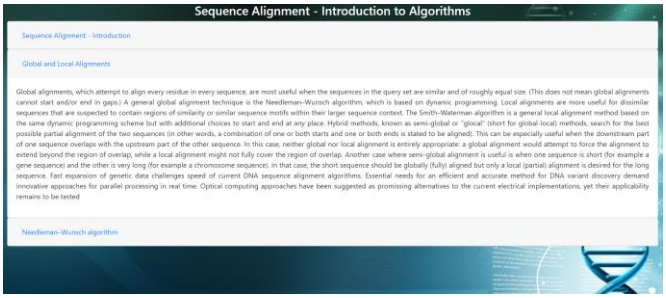
```
function get_mutation(ch){
    var rnum = Math.random()
    if (rnum <= pM){
        var coinflip = Math.random();
        if (coinflip <= 0.5){
                                return get_newch(ch);
                    }
        else
            return null;
    }
 return ch;
}}
```

Probability of mutation is pre-defined and a random number is generated for every letter in the sequence. If that random number is less than or equal to the mutation probability then we flip a coin again, if the random number is less than 0.5, then we mutate the letter otherwise we delete the letter by returning null.

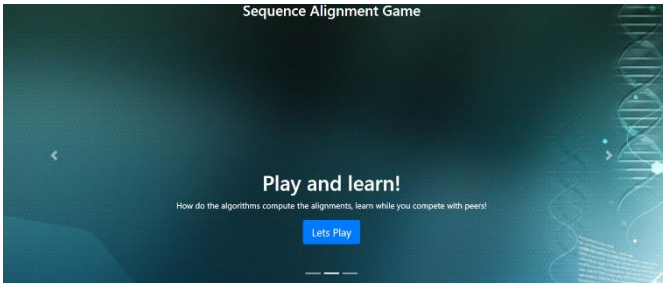
II. GAME DETAILS

The web application is a educational platform which can be used to study about the sequence alignment algorithms. On the home page user shall see a carousal which can be used to select various operations.

You can get information about sequence alignments and the algorithms used to implement them. Right now we have added Needleman-Wunsch algorithm. To read about it you click on learn more!



Second operation is for playing the game. To start the game Click on **Lets Play**!



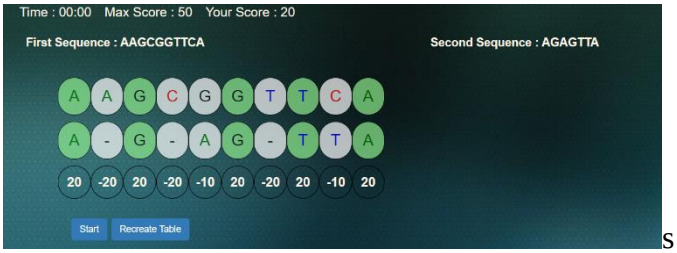
Rules for the game:

1. You get positive points for every match of letters.
2. You get negative points for every mismatch or

gap.

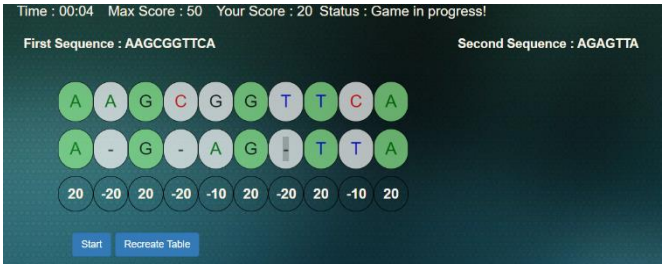
3. Maximum score is already calculated using the Needleman-Wunsch algorithm which you are expected to achieve as soon as possible.
4. Once you reach the final score, game is over, and you can see the time taken using the timer.
5. Multiple users can play this game at the same time

Once user is on the game screen, 2 sequences are randomly generated, and a precomputed max alignment score is calculated for the sequences using Needleman-Wunsch algorithm[4]. This score appears on the top of the screen. Along with this, user is assigned a current score which is calculated based on the current alignment of the sequences.

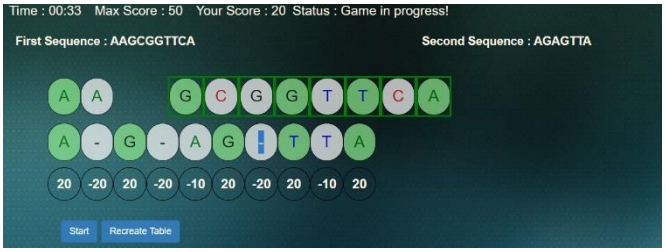


s

Once the user clicks on the start button, the timer on the top left starts ticking and the user is expected to study the sequence and try to align the sequence by either adding gaps or removing them so that the sequence aligns with the maximum score.



To insert a gap user has to pull the sequence toward right and release it.

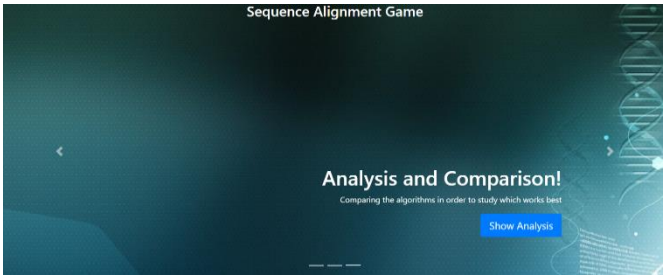


For every match, user can see the letter turning green along with the points it is contributing to the total score.

Once the user achieves the most optimal alignment and his/her score becomes equal to the maximum score calculated by the algorithm the timer stops and user is informed that they have reached the maximum score

If the user wants to associate different values for the match and mismatch of nucleotides, then it is possible by updating the score table. Using the simple scores, the user can play the game with hard coded match mismatch values. Once user updates the scores custom scores can be used to calculate maximum alignment score

In the end, user can view the statistical report of game played and overall performance of the algorithm.



III. EVALUATION - GAME

We asked 22 people to play this game. Out of these 22 people 14 had a background of bioinformatics and have taken at least one course wherein they have studied sequence alignment. 7 others belong to other fields of study and do not know why sequence alignment is done

or implemented.

The objective of this experiment was to find out if the game gives any kind of intuition to the users about how this algorithm works. Our results showed that the game was relatively easier to play for the users who had some prior experience of sequence alignment. However, users from non-bio background showed interest in learning why is this alignment is necessary and what kind of analysis can be done when DNA/RNA sequences are aligned[4].

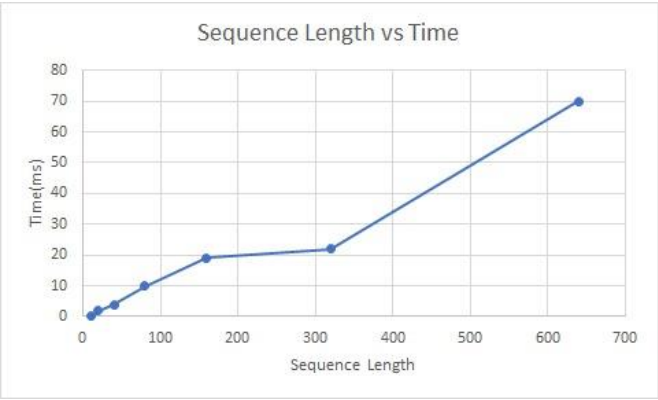
We also received feedback from users that the color scheme for the mismatch should be more intuitive. It should reflect the score that is associated with mismatch and gap scores. Right now, only the match is reflected in the game using "GREEN"

IV. STATISTICAL RESULTS FOR ALGORITHM

Since the game is a very subjective platform to evaluate, we decided to test out algorithm by changing the lengths for the generated sequences. By doing so we observed the relationship between the length of sequence and the time taken to compute the maximum alignment score for the respective sequences.

The following table and graph displays this relationship through a line chart.

Seq. Len	10	20	40	80	160	320	640
Time(ms)	0.1	2	4	10	19	22	70



The graph shows that the time taken to compute maximum score for sequences with longer length grows linearly till a threshold. After this threshold the growth in time taken is exponential.

The time and space complexity of the algorithm is as follows[5]:

1. Time taken to calculate score of one cell in the dynamic table is O(1).
2. Total number of cells in the dynamic table are m*n. Where m and n are lengths of two sequences. So, time complexity of the algorithm is O(m*n).
3. Space taken by one cell is also O(1) which makes the space complexity as O(m*n).
4. It is possible to improve the complexity of this algorithm to O(m*n / logn) by using the method of four Russians.

V. FUTURE SCOPE AND CONCLUSION

The game has a very wide scope for enhanced features. If time permits, we will add more algorithms to the project and create different levels for the game using them. We also plan to work on the feedback we got from our beta users regarding the color scheme and more intuitive UI. Sequence alignment is a very important part of genetic research and hence creating interactive platforms to study these algorithms has considerably high demand.

VI. WORK DIVISION

The project is a product of combined effort of two students –

Sonal Joshi (6540-9104) -

1. Created the interactive front-end of for the game using HTML/CSS and bootstrap.
2. Worked on the real time updates for the score board.
3. Worked on the report for the project.

Aditya Kulkarni (8961-9483) –

1. Implemented the algorithm for calculating the maximum score for the two randomly generated sequences.
2. Implemented the stopwatch for the game.
3. Worked on the report for the project

To review out code you can visit:
https://github.com/comptotherescue/Align-the-Sequence-Game

VII. REFERENCES

[1] https://www.ncbi.nlm.nih.gov/grc/help/
[2] Chakraborty, Angana, and Sanghamitra Bandyopadhyay. "FOGSAA: Fast Optimal Global

Sequence Alignment Algorithm." Scientific reports 3 (2013). DOI: 10.1038/srep01746
[3] Batzoglou, Serafim. "The many faces of sequence alignment." Briefings in bioinformatics 6.1 (2005): 6-22. DOI: 10.1093/bib/6.1.6
[4]https://www.cs.sjsu.edu/~aid/cs152/NeedlemanWunsch.pdf
[5] FN Muhamad “Performance Analysis Of Needleman-Wunsch Algorithm (Global) And Smith-Waterman Algorithm (Local) In Reducing Search Space And Time For Dna Sequence Alignment” et al 2018 J. Phys.: Conf. Ser. 1019 012085
[6] wikipedia.org/wiki/NeedlemanWunsch_algorithm