Linear-Space Local Alignment of The Human and Mouse Titin Protein

Musab Shakeel, Thomas Tarantino

Goal

Our goal was to implement a linear-space complexity local alignment algorithm to locally align the human and mouse Titin proteins.

Background

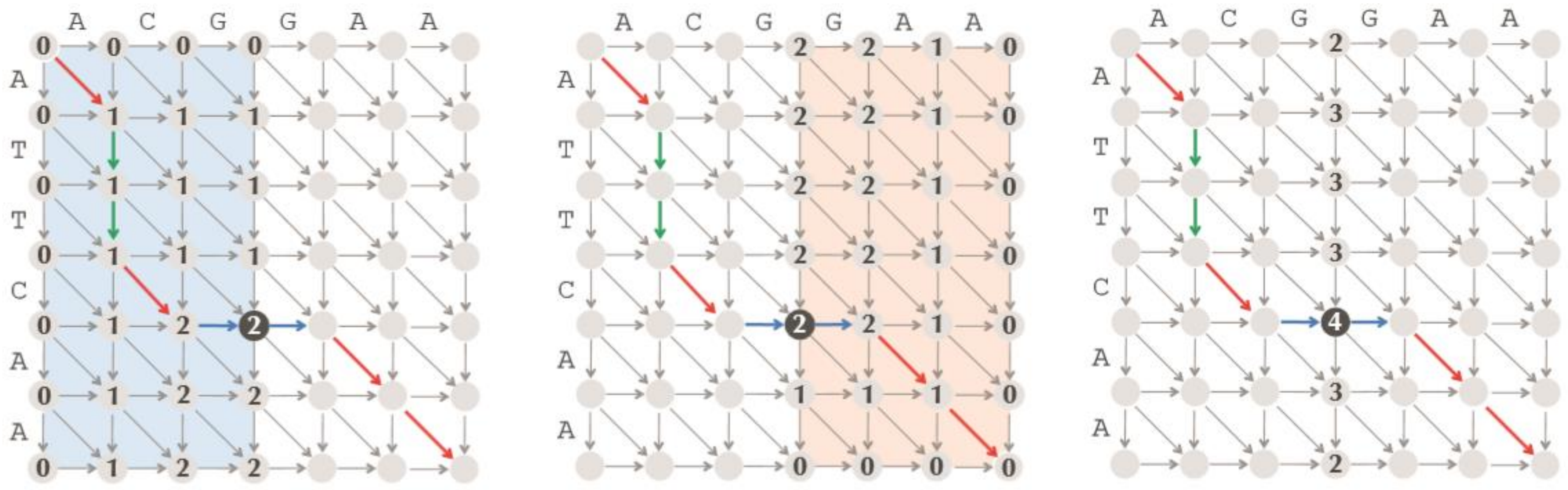
TTN gene is a Protein Coding gene that forms a key component in the functioning of vertebrate striated muscles by providing instructions for making a protein called Titin. Titin plays a vital role in the muscle movement for both skeletal and cardiac muscles. The main job of the protein is to provide stability and flexibility to the muscle cells in order to maintain the balance of forces between the two halves of the sarcomere (the basic structural units of myofibril in striated muscles) as the muscles contract, stretch, or relax. Additionally, Titin has also been found to play a role in the chemical signaling and assembly of new sarcomeres. Mutations in this gene can make muscles unable to contract or relax normally and lead to muscle fatigue, particularly in the shoulders, hips, and limbs.[1]

As our final project, we implemented a linear-space complexity local alignment algorithm to find the optimal local alignment of human and mouse Titin protein sequences. Local alignment allows us to compare the genetic differences and/or similarities in the protein between two species by aligning their sequences most appropriately based on a scoring matrix. Since certain similarity scoring matrices are more effective at certain evolutionary distances, it is important to choose the proper scoring matrix when aligning the human and mouse Titin proteins. Because we expect the sequences to be similar, we used the VTML20, which is a “shallow” scoring matrix, meaning it targets alignments that share 50 – 90% identity, reflecting much less evolutionary change.[2]

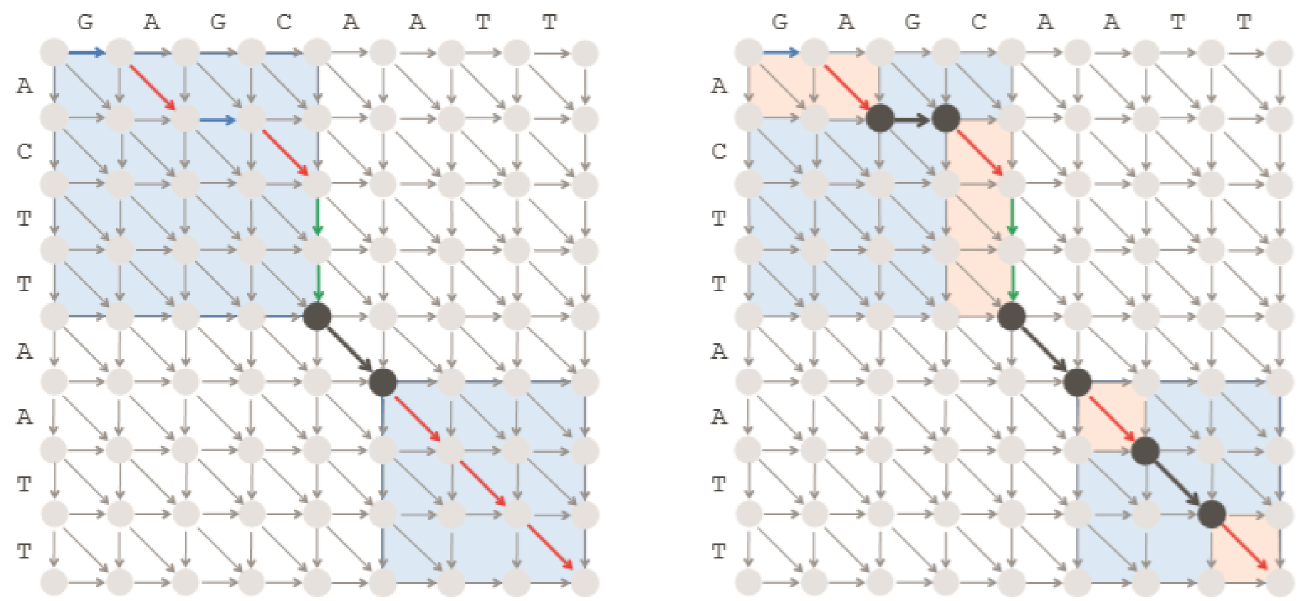
In class we implemented a quadratic-space complexity local alignment algorithm. However, because of the huge memory requirements, the quadratic-space complexity algorithm is not practical for aligning proteins like Titin that are tens of thousands of amino acids long (the "canonical" human Titin protein is 34,350 amino acids long). Therefore, we worked on implementing an algorithm with reduced space complexity so that we could successfully align the Titin protein from humans and mice.

Algorithm Overview

- **GlobalScore:** Compute score matrix for global alignment in linear space (by storing two columns and updating them)
- **Middle:** Call **GlobalScore** on first half and reverse of last half of alignment matrix to find the middle node. Store one column of backtracking pointers when running **Global Score** in reverse – the middle edge is the edge coming out of the middle node



- **LinearSpaceAlignment:** Construct global alignment by calling **Middle** recursively to find all the edges in the longest alignment path



Now if we knew the start and end nodes for local alignment in the alignment graph, we could call **LinearSpaceAlignment** on the substrings of the original strings where the substrings are determined by the coordinates of the start and end nodes. To find the start and end coordinates, we use **LocalScore**:

- **LocalScore:** Computes score matrix for local alignment in linear space while keeping track of the maximum score seen so far, and the coordinates of the node where it occurs. Backtracking will begin at this maximum score node
- **LocalAlign:** First calls **LocalScore** on the two strings to determine the end coordinates. Then calls **LocalScore** on the reverse of the end coordinate-based substrings of the original strings. The starting node should be the node that has the same maximum score in reverse as in the forward direction. Then, **LinearSpaceAlignment** is called on the substrings of the original strings, based off the start and end coordinates just computed

Pseudocode

```
LocalScore(s1, s2, maxscore=0, reverse=False)
    -  Dynamically computes the score matrix for local alignment in linear space by storing two columns at a time.
       Setting optional argument "reverse" to True stops the function when the specified maxScore has been seen.
    -  if reverse == False:
           ○  return maxAll # tuple containing maxScore and the coordinates in the alignment matrix where the
              maxScore was reached. These coordinates form the end coordinates of the local alignment
    -  if reverse == True:
           ○  return startCoord # tuple containing the start coordinates of the local alignment

LocalAlign(s1, s2)
    -  maxAll = LocalScore(s1, s2, reverse=False)
    -  maxScore, endCoord = maxAll
    -  endrow, endcol = endCoord
    -  startCoord = LocalScore( rev( s1[:endcol] ), rev( s2[:endrow] ), maxScore=maxScore, reverse=True)
    -  startCoord = ( len( s2[:endrow] ) - startCoord[0], len (s1[:endcol] ) - startCoord[1] )
    -  startrow, startcol = startCoord
    -  aligned = LinearSpaceAlignment( s1[startcol:endcol], s2[startrow:endrow] )
```

Experiment

We ran our linear-space alignment algorithm to align the human and mouse Titin proteins, which are 34,350 and 35,213 amino acids long respectively. Our program took ~ 3.5 hours to finish and the resulting local alignment had a score of 221,457. This score is not meaningful without context so we looked at the length of the alignment which was 35,286. This is longer than either one of the protein sequences. This indicates that that Titin protein sequence is highly conserved between humans and mice.

Titin (human) Length	Titin (mouse) Length	Local Alignment Length	Local Alignment Score	Total Time Taken
34,350	35,213	35,286	221,457	3.5 hours

It would be interesting to use the linear-space algorithm to align Titin from humans and from other mammals and compare the score from the respective alignments to determine the mammal whose Titin is most similar to the human Titin.

References

1. Chauveau, Claire, John Rowell, and Ana Ferreiro. "A rising titan: TTN review and mutation update." *Human mutation* 35.9 (2014): 1046-1059.

2. Pearson, William R. “Selecting the Right Similarity-Scoring Matrix.” *Current Protocols in Bioinformatics*, U.S. National Library of Medicine, 2013, www.ncbi.nlm.nih.gov/pmc/articles/PMC3848038/.

3. P. Compeau and P. Pevzner, “Bioinformatics Algorithms: An Active Learning Approach,” 1st ed., ch. 5, pp. 240-285, 2015

Acknowledgments: Our project would not have been possible without the guidance and continuous support from Professor Linderman