



T.C.
MARMARA UNIVERSITY
FACULTY of ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

Computational Genomics Project- 2

Ulaş Deniz Işık – 150118887

Bedirhan Sarıhan – 150119692

Merve Hazal Özalp – 150120827

In our code, we implemented the same recursive function calls. We kept 3 arrays which are lowerMax, middleMax and upperMax. We first call the middle function with passing the length of 2 alignments. After function calls ended, the arrays will be filled. Then we traceback. Every position in these arrays shows us the score we get if we came from that way. For example, for a random position in these 2-d arrays, if the middleMax array has higher score than the other arrays for this position, it means that for the optimal solution, we came this current position from the diagonal position.

We tested our score result by comparing them with a python tool known as Biopython. After we get the same optimal results, we started to traceback the solution. When we get alignments, we have to make sure that it really gives the same optimal score. It is possible that we get the score optimal but the sequence does not give the same score due to a logical error in our traceback function. Thus, we wrote a function to test the score of the alignment we produce. This code is not in our source code because we use this function only for testing.

```
function alignmentScore(seq1, seq2) {
  let score = 0;

  for (let i = 0; i < seq1.length; i++) {
    // If the characters at the current position are a match, increment the score by 3
    if (seq1[i] === seq2[i]) {
      score += 3;
    }
    // If the characters at the current position are a mismatch, decrement the score by 1
    else if (seq1[i] !== "-" && seq2[i] !== "-") {
      score -= 1;
    }
    // If either character at the current position is a gap, decrement the score by 1 for
    //the gap opening penalty and by 0.5 for the gap extension penalty
    else {
      if((seq1[i - 1] === "-" && seq1[i] === "-" ) || (seq2[i - 1] === "-" && seq2[i] === "-")){
        score -= 0.5;
      } else {
        score -= 1;
      }
    }
  }
  return score;
}
```

If current characters are the same, add +3 to the score. If they don't match and there is no indel, decrement score by 1. If there is an indel and its the first one (meaning that it is a gap opening), decrement by 1. If there is an indel but it has also an indel on the previous position on the same sequence, decrement the score by 0.5.

We get the same optimal results. Now we can say that our code is working properly.

Output for test1.seq file can be seen below:

Alignment 1:

```
C--GAGACCGA-CG-AAGAGGTTTG---GCCCAAC-CAGGTTCC-CTGATCACGTAACCTACCGGCCAAA-----AGGACTG--GCCTTA-CTAAGGCC--TTT----G
TCTACTGC-G-G--GT-C--CGG-G-GGC-CGTT--GGT-T-TCGGCAGAAC-ACTTC
```

Alignment 2:

```
CCTG-GACCGAGCTTAA-A---TTGCTAGC---AATACAGATGCCGCT--TC-----CTT---GGGGAGGGTGTGTAGGA-TGTAGG-TTAACGAATGCAAGTTCGGGG
TAT-C-GCAGAGTCGTGCTACGGCGTGGCAC-TTAGGGTCTCTCGGAAAAAGAGTAG
```

Score: 189

Output for test2.seq file can be seen below:

Alignment 1:

```
GTGT-GGTTGCTTGCACTCCGTGTACATGTGACAACCGAAGAGTTGATCCAGCTTTGTTAAGTCAGCTTCGAATG-CGGTAGCTCTCAAA-TATGAT-ATGA-C-TC
TTGGGGTAGAT--GCTGG--GGACCTATTGCGCCC--AAAGCGAT-----ATTCGGGGCA-CCGGTTTAGGGTACCCTATCAAGGCGAT-AC-TTCGATGCAGTGTG-ATG
CCGG-A-G-GTGTGC-GTCAGCATGTGAGA--GCT
```

Alignment 2:

```
G-GTAGGTT-CGTGAAGCACTCC-TGGAC-TCTGACCAC--AAC-A--TAATCAAGC--GC--AG--AG-TT-GAATGACCAAAGCGCTCAAGCT-T--TCACGAACGTC
TC-----ATATCCGC-GGTCGTACAAAC-GCGCTCTTAAACAATCGTCTAT-C-----CATCCGG-----GGATACCA-AT---GG-G-TGACATT--A---ACTGGGCAGG
CCGGCACGCG-GACGCGACAG-AT-TGAAATGGAT
```

Score: 342

Output for test3.seq file can be seen below:

Alignment 1:

```
CGGG-GAAAGAC-GGA---ATGCATCGACCA-TCGGACAAT-GCCTC-ACTGGAAGCGCTG--CTGATTTTTCGCGAA-CGAGCCTCGGA-CCTCCCG-C-TCAAACCT--
ACGAAA-AT--GACTCCACC-----AGC-ACTGAAC-CACTGGCCTCCAG-TGA-AGATA---GTATC-TACAAATCGTTTGCCGGGAGAAACATT-TGTA-TGGTAG-T
CAGCGTTCTGCACGTCAC-----GTAATCGT--TCACTAGTT-GTGG-GGTA---TACCAGGTC---GTAATGAG-----A-TG--TTAGTA---TGAATCGTTTATAACG
-CTTGTCATAGGAGTT-C---C-GAATAATCGTCACT-AGGTCAA-T-G---GCCCCCTACT-TGTA--ATAACTACGT-CTGATTGAGAAA----CAGATCGTTAGTCA-
TCGGTT-AATAGTCCCGAAGATAACGGGTTCTTGCGTTTTTGCGAATACTACTATC-TCATGGCGATGG--AG----C-GT---T-TGG---TCCCAT-CC---AGCCGC
GC---GAGTATCACTTGTTGCGCTGCA-C--CTG-TCCGACCTTTCGAT-----GGGGA---GCTCCTTTCATTGGTT--GTAG-GTACT-AAGGGTGAGCAATGTCACGT
GACCCAA-GGAGCC-GTGTGTAATTTCCACTTGCTCAGAAAAG--CCTC-GACA----AT--CTG-GGA--CCGACACCTGAGTGA-T--GGCTTACA-GACCAGGGGAGG
GG-G-GCAGGTTCCCTGGCCACAGAATGGCACGCCCTGAGGAGGCACCGGCCCAA-CGTGCG-TGG
```

Alignment 2:

```
AGTGAGAA-GACCGGATATATGCAACGAACAATCGCAAAATAGC-TCCA-TGTACGC-CTGTCCCG-TAATACCGCATGCGAGCCTCGAAGCC-CCCGACGTCAAACCTCA
AC-AAGTATAGGGTTCCACCGTATGAGCCACAGAATTCAGTT--CCTCC-GCTGATAG-TCCCGGT-TCCTCAA-TCA---GC----AGA----TTCTG-ACT--TAGGT
CA-----CTGGAAGAAACCTGGCGTATT-GTGATGAA-ATTTCTGGTGG-ACGCTACC--GTCTAGGTA--GAGCCACTAGTGCATT-GTACACTG--TCTTTT-TC-CG
GCTA-TAATCGGAGTTTCAGGCTGACT--T-GCCACCCAGGTCAAATAGTATGTC----ACGGTGATGATCTCT-CGGGCTGAT--AGCAAGGGGCGG--CGTCGGGCAC
TCC-TTGAATTGACC-----ATTT--GG--CGTGCCCTTGTTGC-----CTCCT-TCCTCAAG-C-AGGGTTAGGCTTCAGTAAGTGTGGGGGTGGC-TGCCGAAAGACGG
GCTCGGGGT-----GTT-GCC--CAACGTCTGGTC-GAC-TTCCATAATCTGGGCACAAGA--CGT--AT-GGACCCGT-GCG--CTTAATGGTG--CATT-TCACGC
GAGA-AACGGAGGCAG-GTGCCAT--CCGC-GC---GAAAAGATCATGACATGACATAAC-GAGGAACCCGAT-CCGGAGTGAATACGTC--ACATG-CCAGGGG-GC
GGTGTGCAG-TTCTC-GACCG--GAAACGCCACCC--ACGACGTACCG-CCAAGCGTCGCTGG
```

Score: 1125.5

Output for test4.seq file can be seen below:

```
Alignment 1:
AGGC--CGAAAACGTCGCGAAT-T-GACCCTGGCGACGCCGCGAACGGGACCTCCGTTAGT-GTG----GGAGGTCATCAATCTCGTTCG---CTAGCGGCTGACACCAA
TCACTATAAGTCTGTCATGAC

Alignment 2:
---CTTC---AA-GTCA--AATATAGATCCTGGC--CGCT-CC--ACGGG--CT---TAAGTCGTTCTCCGAAGGT-A-CGATCTGGTTGGATGCTTCCGTCTAA-AC-AA
GAAG-ATAA-----TC--G--

Score: 163
```

Output for test5.seq file can be seen below:

```
Alignment 1:
CGG--GTAGTTAACCTA-CAG-CATAGAGTCGCGAGATAAAGTGCAGGA-GTCTTTCGCGGCAGATTCTGTACCTCAACCA-CGT-GCTACTT---TCTGGCATCACGAAT
CTGCCGCATAGGTCCGTGAGT-CCATATGA

Alignment 2:
AGGAAGTAGTTAGCC-TAACAGGCATAGAGTCGCGACATAT-GTGAAG-ATGTCATTGC--GTA--TTCAAACCTCATGCATCATTGC--CTTGAGTC--GC-TC-----
CTGGAGCATAG-TCCCTGAGTGCCATATGA

Score: 249
```

To run our code, you have to follow these steps:

- 1- Install nodejs to your computer.
- 2- Open the terminal and navigate to project folder.
- 3- Run “node index.js” command.

File path is default “./test1.seq”. The FILEPATH variable is defined in the 2. line of the code. You can change it to execute different file.