Megan Chu
A12814536
Assignment 2
4/25/18

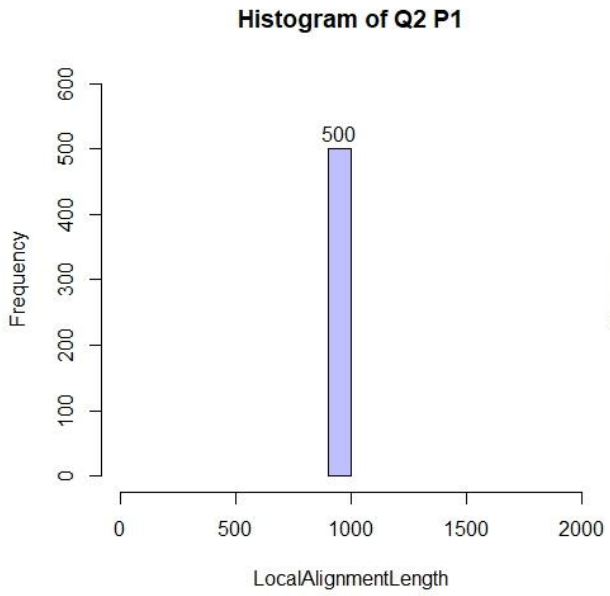
Q1. **Program Output:**

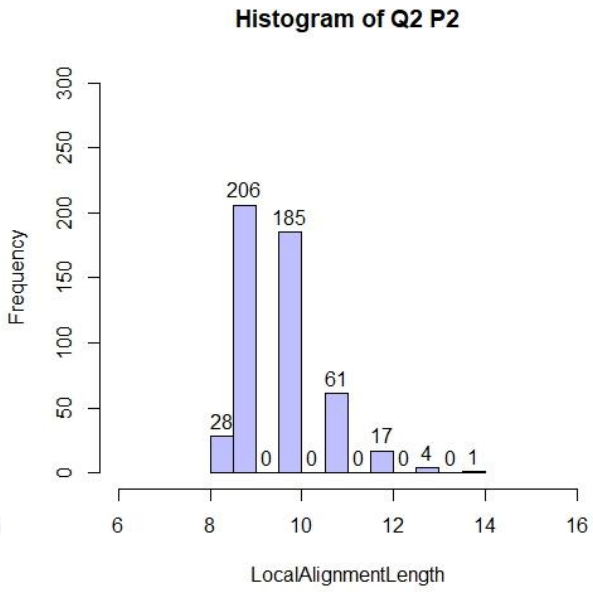
```
>python Q1.py p1seqs.txt -m 1 -s -10 -d -1
Running Q1(locAL)...
Score of the best local-alignment: 119
*Note there may be many local-alignments with the best score, this
code only outputs one of the local-alignments with the best score.*
Length of one best local-alignment: 1049

>python Q1.py p1seqs.txt -m 1 -s -10 -d -1 -a
Running Q1(locAL)...
Score of the best local-alignment: 119
*Note there may be many local-alignments with the best score, this
code only outputs one of the local-alignments with the best score.*
Length of one best local-alignment: 1049
Actual local-alignment:
239 7
ACAAT-A--A-CACCT--AAGAGC--AAAA-TCAAT-A-AA-ACACCGATCCTCCGAGG-A-T-
AACCAAG-AGAGACCTAAGAACGACAA--GAAACCAATGAA-AG--A--A--AAAGA-AAATGG-
ACATCA-GAACGA-C---T-TAGAATGCT---GGGAAAA-AG-AAA----
AATTATAAACGAAGGATGGGC-ATAAATT-GGACG-A--AG--CC-AAG-AGATA-GG----CCGA-
GAT-A-A-A-ACGG--AG--AAC-AA-T--AA--GG-G---AG-ACCAT--GGAGAGCAAACC--A-
ACCGCAACAAATAA-----A--GG-GGGGGACAA--AAAC-A-A-G-ACC-
AACCCAAACTGTCAGACAGG-A-A---GAG-
CAATAACCAAGACAGAAGAAGAAACAGGAGACAAACAACATAATATAAGAGCACC-TAG-CT-AAC---
AA-AAAA--GA-CC-AGCAAACGGAT--T--AA-GA--AGATA--A-AG-A-AA-ACGT-AAAG-A-A-
CAGTCA-AG--GAACAAG-CG-AT-AATA-AATGCAGGGAAA--AAATGGGGACAGACGAAGGA--
AACAACC--A-GAA--ATAA--TCTAACG-CATCGCAGAA---GATG-AC-ACTG-
CGAGAAAATACGAGCC-GTATACGACACAA-----AACCGGG-AATAA-AGA--AAA--A--AA-C-
CAT-A---C--CC-A----A-A--A----A--GA-ACA-----AC--GCGAAAGATGA-A-ACGCT--
CCCAACTCGGATGA-G--CAAAG--CCGCCAGGCC-A-AAAAA----GAGAA--CC--A-GAG-C-AG-
AG-C--G---A-AG-CT-AT-GG-G-T-AG----A---AA-ACAC-CC---TAA-GCGC---
GGGTAGTAG-A-G-ACG-AAA-A--AT-AA-AAACAG-G-CT--G----AC--CCG-AACATA-AGAG-
C-C----CA--C---ACA-A-G-T---AGA-AG--A-AC--GGAAAGAAAACGAAAAGA
AATCACACCACAAA--GGAAAGA-TAAGC-GCAAA-GAGAAGT----AC-CC-T----
GCATACCTACAA-CC----AA-AT-AAA-A-GAGGAGAA--C--TGAGAAACGCCACCAAA-ACAAA---
CGTGA-CGATA-ACTAATGAAACGAA-T-GAAAA--AGGAGGATAGACCTCAA--A-TT----C--AAA-
-GAGATGAA-CATGACAGCTAAAAG-ACAACGAG--CAAAAATG-C-TAG-GAGACATA-
ACCAAGCTAAAGACCAGGACCCAACGACCGAACGCA-AGAAT-GA-AAA--TTAGCCC-C--AAAA-
AACGCGCACGAGAAA---G-AACTAAA-GAGCCACAAAGACA--C-AA---TA-G-AA-AGTGCTCG-
ACGG--ACA-AA-A-A-AA-AA-AAA-GA-A-G-AG--ACAC------AAATAGAA-ACAAAA-
ACAAAAGAAGCAAACGAATA-TGA-C-AAAGA-AAACCAACTACCA-AAAGCAGTACATGACAC--
ATCA-TGCACACA-GCGAAAACAACAA-AA-TGAACA-ACGA-A-AA--CAACCAAA-----A-A-G-
AC-GAATCG-AACA-GGAGAGAGGA-T-CCCC--AA-AAA-A-GAGG-GCCCA-ACTAAGACAAT-GC--
AA-A--GA--AA-C-GCG-ACAAAGCCTCG-CCAACAGA-ATCA-ACCAAAGCATGAACAGCA-
CTTTTAAAACATGTGGCGCGGCGTCGAGCAGTA-CGGTTCAAATGCAAAA-A--TTACAA---AAA--
AG--AC-A-TGCACTGAAC-CCCGTAAA--
GAACGAGAAACTTCAAGAAGGAAGAAGACATAAGCCAAAAAACCAAATAATAGACACAGCTCGAGAAAAA
GCCCAACAAACGACAGAAAAGAAGGGGAGTAAGGAGAAGCAAAACAGAACAGGGGCGAGAACGACTGCGG
```

AGTAATCGAAAGACATGACAAGTAAACGAGAAAAGATTAATAAACGACATACACAAACAAAAACAATAGA
AACGAAAAAAAAAAAA
107 14

*Note '237 7' in the output above means 237 bp were cut off from the beginning of seq1 and 7 bp were cut off from the end of seq1. Similarly, '107 14' in the output above means 107 bp were cut off from the beginning of seq2 and 14 bp were cut off from the end of seq2

Q2.
Note I put complete program output of Q2 as a text file named Q2randomSeq.txt and included it in my submitted code.
End of Program Output:
...
CTATGCACAGGTGAGTCGCCGTTGTCGCCTTAACGCATGTTTCTACGTCTACTTGGACCGGTCGCATTAG
GACCCTAACCAACCCTTCCGCTCTTGCATCTGTAGCCGCGTTATTTTGAAATTTAATCACACGGGTAACG
CCATCACGAATGTAGCTTACCTGTCAGCTAGTTGGAATGACTGAGGTCATCACCAGACTGATATGTATTA
TATAGAAGTGAGCTGCGCGCATCAGCCGCCAATATCCAGATGATACCTATTAACCTCAGCTAACGGGCGT
TCCTTAGGATATTATCGTGGTCGCCCTGGTAAAAGTCCAACGAGAAACCTCCTGGGATATCATCTATAAC
GCCGTTATCATACCGTGGACTGATCGTCGGCAGATCAGCGTCTCATGATAAGTCTGCCAATTTCAATCAG
ACTGATACCATTGTCGCAACGAAAGCTAACTCCGACAAGAGTGACTAAAGCGATTTATAATCGGTCACAA
GGAGTCAACACCGCATATGTAGTCCGAGAGCGCTACTCCTCAGGGTGAAAATGCCTATAATTACCCCGGG
TTCGAGACCCGGTGGTGACTCTGGTCTGAGCCAGCTATGATCTCAATGATGGGCTAATTATTCGGGAGCC
AAATGAGACCCCCACATAGGTATATATGTGCCATGATAGCCGTCCTTTGCGTGGGCTGAGCCCCATCTTT
ACGTAGCGTGCCCCGTTCAAGTATCTGGTTCTGTTTGTTCATGCTACTTCTTCATTGATGTGGCTGAACA
ATAGGGCGCGTGACGTAGATTCGCCGCGATATGTTTAAACTCCAATTCCACTATACTCCTATAATAGGCA
GATCATAATTCCTAAGATACGCTGATGCTACACTTCGTAATTTGTTACCTGTAGGGATAATCCAGCGGGG
AATCCTTTTCATTTATTTGCTGTTATTAGTACACTAAGCGACAAAGCAGGGATATTACTATCCCCCATCT
CGCCATTGTAATACGGCAAA
CGGTTGGCGACACCAACGTCGATTGGGACCCTGATACGGGTCCGTTCCTGTTAGAGGAGCGGTCAAAGCA
TTTTAATAAGTTAGGTACACACGTAAGTTATCCTGCAGCGCTAAAGGTCTGAGCCATGGCCGTGGGCCCA
ACAGTAGCTAACTAATAAGGTTCCCCAAGGAGGGCAGGAATTAACCCGGTCGCGGGGCCTCTACTCTGTC
CATCCCCATTATCAGTGCGTGTGTTCTAACGATCTGCATCATGGAGTAGGCCCGTGCCGTATCGAGTCAG
CTGTGTCTGTCGGATGGTACAGACCAGCACTTGGCTGAGTATACGTGGAGACTTCAACTCGTCTCATCAG
AGTTTTATTGCCACATACGCACGAGCGTACTGACTGACCATGCCCGTCATGAACCTATGTCCTGAAAAGA
CTTAGACTCATGCCAGGGTGGTTGCACCGAATGCCATGTTCTTGTGTTAAACGTGAACGCAGCCTGGCTG
GAAAACCAATGTTTTGATGTGGTTTTTAAACGATTCTAAGAGGAGAAGAGGGCCTGACACTAGGACGACG
AAAGCTCAGACCCTTCATCGCTGACACCACGATGGGGCACTGTTATCCGATGATAGCCGAAGGGGCACTT
TTCCCGTCGTTCAATTAAGTTCCGGTGAGAGATGAAAAGCCTAGCACTGTGTCCAATCCCCGTTGATCAC
CGAATTGCAAATAGTGATCTAGGGTAGCCCCTTAGGGAGTCTACTCGTGGCAATTGAGCCGAAAACACCG
CTCGTTCTAGATTGGAGCCTATTTACGTGGTGTCTCCCCGAGAAATTCGCGGCTCTGTCAGAAGGGTCGA
TGTTGCCGCCCACATCCTTATCATTCCCGTGATGATAATCCAGTAGAGTCCAGTGGTCACTTTGTAACGC
CACCCTAGCAACTAATTCGTGCCTACATGCGCCTCGCACATCGGGCACTTACTACCACAATATTATCTAA
CTCTGTGTCCTATACTCCTC
Frequency A: 0.25
Frequency C: 0.25
Frequency G: 0.25
Frequency T: 0.25





Note in the P2 Histogram, the first column with 28 should represents length 8, the next non-zero column (206) represents length 9, 185 represents length 10, and so on…

Q2 Answers to questions:

I calculated $l_{P1}(n)$ using n values of 1, 10, 100, an 1000 and using 50 pairs of random sequences (instead of 500 pairs to speed up runtime). The expected length of the local-alignment mostly equaled the n-value or length of the initial sequences for low n and almost always equaled the n-value for n >= 100. This leads me to guess that $l_{P1}(n) = n$.

I calculated $l_{P2}(n)$ using n values of 1, 10, 100, an 1000 and using 50 pairs of random sequences (instead of 500 pairs to speed up runtime). The expected length of the local-alignment always hovered around the cube root of the n-value or length of the initial sequences. After more careful observation of the values, I figured out the n values and $l_{P2}(n)$ followed more of a logarithmic relationship. $Log_{10}n$ gave local-alignment lengths that were smaller than the observed. Trying log_2n I saw that the values were very close to the mean $l_{P2}(n)$ values. This leads me to guess that $l_{P2}(n) = log_2n$.

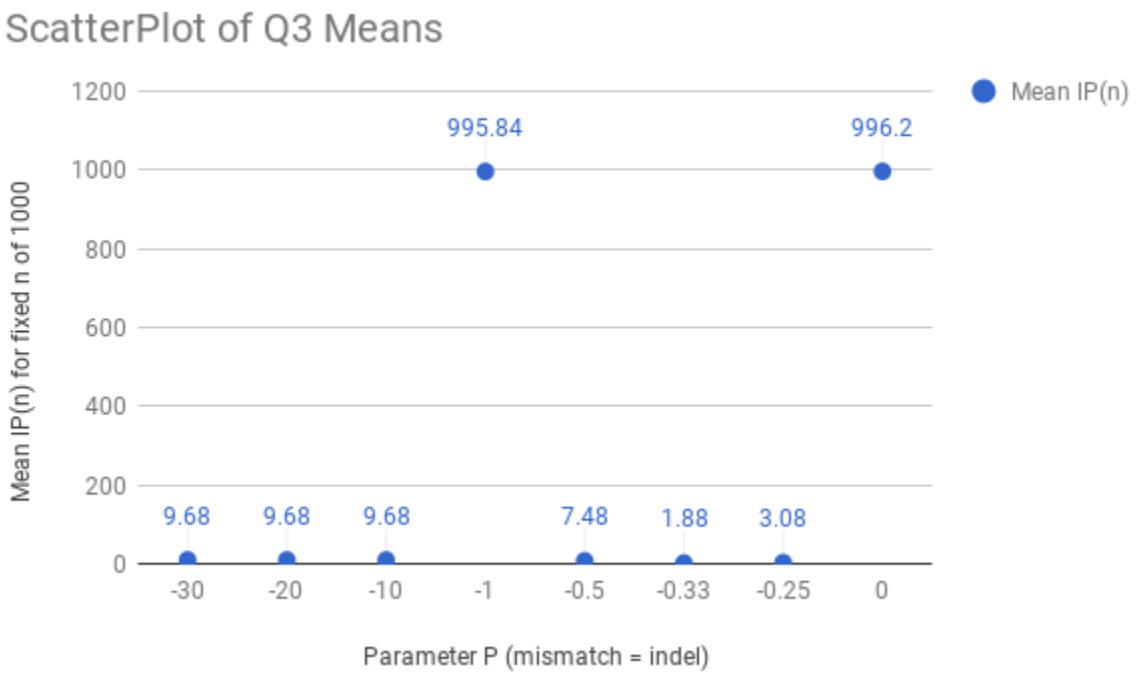
Q2 Extra Credit Attempt:

$l_{P1}(n) = n$ makes sense since we don't get penalized for indels. But we get heavily penalized for mismatches. This leads us to believe that any local-alignment for P1 will have little to no mismatches, and will contain mostly matches or indels. We see that we can place indels in sequence 1 such that we match every base, however sequence 2 will not be long enough to match every base in sequence 1 (since seq1 is now longer if we add indels). Thus the local-alignment length is bounded

by the length of sequence 2 (the length of the initial sequences without addin indels), which is n.

$l_{P2}(n) = \log_2 n$ makes sense since we are heavily penalized for both mismatches and indels. We see that indels are slightly less penalized than mismatches. This leads us to believe that any local-alignment for P2 will have little to no mismatches, maybe a couple indels, and mostly matches. We see that our local alignment cant start or end with an indel/mismatch because we can just cut off the indel/mismatch and get +20/+30 to our best score, respectively. Thus, any indels/mismatches must be inside the local alignment. Since it is highly unlikely to choose a mismatch if we can use indels that have less penalties, let's say our local alignment contains matches with a few indels. We see we need more than 20 matches in a row for an indel to be favorable, and for the final local alignment to contain the indel, there must be more than 20 matches in a row after it. This is highly unlikely, so we might as well only look at local alignments which contain only matches. Now the probability that we have a match given a position in seq1 and seq2 is ¼ (if we know seq1 base). We see that the probability of having x matches in a row is (1/4)^x. We have n-x+1 such sets of x bases in a row given sequences of length n. Thus the probability of having x matches in a row given sequences of length n is (n-x+1)(1/4)^x. We see this equation returns higher probabilities when x is small. Since $x \sim l_P(n)$, and as n increases, higher valued x will give a higher probability, and this probability increases on a logarithmic scale. Thus, $l_{P2}(n)$ must relate to n on a logarithmic scale such as $\log_2 n$ if not some scalar of that.

Q3. **Program Output:**
a) I followed the suggestions, set n = 1000, and got average $l_P(n)$ after running for mismatch=indel for {-30,-20,-10,-1,-0.5,-0.33,-0.25,0}



b) Yes, there is an abrupt change in the value of $l_p(n)$ when mismatch penalty = indel penalty = -1 and when mismatch penalty = indel penalty = 0.

Q3 Extra Credit/Research Attempt:
I believe that the likeliest reason we get $l_P(n)$ lengths of ~ 1000 when mismatch and indel penalties = -1 is because we set the "reward" for a match as 1. Suppose our local-alignment scoring matrix had a 0 at some coordinate after a mismatch/indel penalty. We see that we only need 1 match to keep the next alignment score over 1. What I am saying is that we can have a local alignment as shown below
ACC-GT-
ACCC-TG
As long as we have 1 more match than mismatches/indels in the alignment the alignment can keep on going. This is much more likely to happen than if we had high mismatch/indel penalties like -30, -20, or -10, where we would need 31, 21, or 11 matches per mismatch.indel in order for the alignment to continue. This is why at mismatch penalty = indel penalty = -1 and match = 1, we can have a much longer local alignment, because the chances of keeping any score in the alignment matrix over 0 is much more likely.

In the other case where mismatch = indel = 0, it is kind of obvious that we will get a local alignment with length close to n = 1000, the lengths of the original two strings. We see that the best scoring

local alignment is given by one where all bases in seq1 or seq2 are matched. Since indels and mismatches are not penalized we are just competing for the most matches. In this case, we can just keep inserting indels in one sequence to get a matches for all its bases. However, the other sequence does not have enough bases to match EVERY base in the first sequence if we keep indeling the first sequence. Thus we see that the length of the local alignment depends on the length of the initial sequences (the length of one sequence without indels), which is ~1000.

Q4. **Program Output:**

```
>python Q4.py p4seqs.txt -m 1 -s -10 -d -1
Running Q4(locAL)...
Score of the best local-alignment: 56
*Note there may be many local-alignments with the best score, this
code only outputs one of the local-alignments with the best score.*
Length of one best local-alignment: 78

>python Q4.py p4seqs.txt -m 1 -s -10 -d -1 -a
Running Q4(locAL)...
Score of the best local-alignment: 56
*Note there may be many local-alignments with the best score, this
code only outputs one of the local-alignments with the best score.*
Length of one best local-alignment: 78
Actual local-alignment:
21156 3774
CGTTATGA--GCGATG-G---G--
ACCTCATCCCGGATTAGGATACTTCACGCTTACGAACTCTCAGGGACAGTTCCG
GA-GTGG-CTGAGC-
GCTTTCGACCTCATCCCGGATTAGGATACTTCACGCTTACGAACTCTCAGGGACAGAGTTCCG
3572 21353
```

*Note '21156 3774' in the output above means 21156 bp were cut off from the beginning of seq1 and 3774 bp were cut off from the end of seq1. Similarly, '3572 21353' in the output above means 3572 bp were cut off from the beginning of seq2 and 21353 bp were cut off from the end of seq2.

Q5.

I used python for the assignment, and I used R to plot the graphs. I spent around 12 hours on the assignment. I did not discuss my homework with anyone, but I did look at other peoples' questions on piazza for clarifications on the homework instructions.