CSCI 291T (Spring 2015) Programming assignment 5 25pts Due: March 12 (Th)

Task: median string and motif finding

Write a program to find the median string and motif from given a set of DNA sequences and the length of the motif. Your program should report the best 5 median strings and their corresponding motif consensus strings and positions.

Input: a set of DNA sequences (arbitrary length each) and motif length (L-mer);
a prokaryotic DNA sequence file "HMP-617.fa" is placed in the Blackboard.

Output: best 5 median strings (with total_distances), corresponding motif consensus strings (with consensus scores), and motif positions (for each motif string);
refer to the sample output shown in the next page.

Suggested approach:

For the task, you should start with finding the median string and then, find the motif consensus string based on the median string.
The benefit of this approach, i.e., finding median and then motif, is that the searching space of finding the motif is reduced.

Suggested steps of operations:

1. read input fasta sequence file (arbitrary number of sequences and arbitrary length each) and store that in a data structure;

2. find the median string; using a priority_queue of size 5, keep the best 5 median strings and their total_distances; searching space is K^L, e.g., with 6-mer and DNA data, 4^6=4096 choices; in this assignment, we do not use any optimization technique to reduce the searching space.

3. using the best 5 median strings, find the corresponding motif consensus strings/scores/positions; The original appearing order of the DNA sequences should be kept in the reported positions, i.e., s = (s1,s2,..,st), where s1 is the starting position of the motif in DAN seq1, s2 is the starting position of the motif in DNA seq2, and so on;

Sample input and output are shown in the next page.

Submission:

1. Include good documentations in the source code and submit the hard copy of the source code. Documentation should include the global documentation (at least, program description – what it does, input/output description, methods/techniques/algorithms used, how to compile and run) and each func head documentation (what the func does, methods/algorithms used, input/output).
2. Run your program with HMP-617.fa and 6-mer, and submit the hard copy of the run time output.

Sample input:

```
>ECSE_P6-0003;coding;x      HMP.18057.AP009246.3296.3550.+
ATGAGCAGAACACTCGAACAGAAGATTGCTGATGCTGAGGCCAGATTACAGCGCCTGAAGGCGAAGAGCAGGAGT
CTGGACACAGCGCAGAAGGTCATTGTGGGTGCAGCATTGCTGGCAAAGGTCAGAAAGCCGGAGGAGGTGCAGTTG
CGTGCCTGGTTACTCCAGTTCCTGAAGGCAGAGGTGACACGACAGGCTGATGTGACGCGCATACTACCGCTGATT
AACGAGCTGGAAGCGCTGCCAGGACAGTGA
>ECSE_P5-0006;coding;x      HMP.18057.AP009245.4367.4621.+
TTGAGCGTCAGATTTCGTGATGCTTGTCAGGGGGGCGGAGCCTATGGAAAAACGGCTTTGCCGTGGCCTTATCGC
TTCCCTGCTAAGTTTCTTCCTGGCATCTTCCAGGATATTTCCGCCCCGCCAGTAAGCCGGTACCGCTCGCCGCAG
CCGAACGACCGAGCGCAGCGAGTCAGTGAGCGAGGAAGCGGAATATATCCTGTATCGCATATTCTGCTGACGCGC
CTGTGCCGCCTTTTTTCTCCTGTCACATGA
>ECSE_P5-0007;coding;x      HMP.18057.AP009245.4798.4989.-
GTGACTAAACAGGAAAAAACCGCCCTTAACATGGCCCGCTTTATCAGAAGCCAGACATTAACGCTTCTGGAGAAA
CTCAACGAGCTGGACGCGGACGAACAGGCAGATATATGTGAATCGCTTCACGACCACGCCGATGAGCTTTACCGC
AGCTGCCTCTCACGCTTCGGGGATGACGGTGAAAACATGTAA
>ECSE_P3-0027;coding;x      HMP.18057.AP009243.24565.24741.+
GTGGCGGTGATTATTAACCATAAAATAAAACGCCTACATCTTGAAAATTCTGCCCCCCGTTACTTTATTCGTACC
CCTTATAATGGGGTGTTAGCCAGCCAGACCCGGCATGATTACTGCCCCCAGTCGTCCATGATCCGGGGGGTGATG
TCACCGGGTCTGGTGGGGCGCTGGTAA
>ECSE_P3-0030;coding;x      HMP.18057.AP009243.27049.27315.+
TTGTTTATATCTTATTTTGATTTGTATCTAATAACTGTTTCTTTTGTTTTTATTGTTTTTGTTTTGCGTTTCAGT
GAAATAGCGATCTTTTATTTGTGTATTTTTTCTGTTTTTGGTTGTTTGTGCTGTGCTTTTGTGTTGTTGTGGTTT
TTTGTTCTACAGATTTTGTTTAATTCGCTATTAAAGATCGTCAAGTGCGAGATTGCCGATATAAAAGATCGAAGT
AAAATAGTTATCGAAGCGTCGTTGTTGTGTTTCTTTGATTAA
```

Sample output (with 6-mer):

median string: CAGTGA (tot_dist = 2)
motif consensus string: CAGTGA (consensus_score = 28)
motif positions/string s=(s1..st):
  249(CAGTGA), 173(CAGTGA), 176(CGGTGA), 4(CGGTGA), 71(CAGTGA)

median string: GACGCG (tot_dist = 2)
motif consensus string: GACGCG (consensus_score = 28)
motif positions/string s=(s1..st):
  203(GACGCG), 218(GACGCG), 87(GACGCG), 101(GACCCG), 237(GAAGCG)

median string: GCCAGA (tot_dist = 2)
motif consensus string: GCCAGA (consensus_score = 28)
motif positions/string s=(s1..st):
  39(GCCAGA), 6(GTCAGA), 49(GCCAGA), 97(GCCAGA), 196(GCGAGA)

median string: ATTAAC (tot_dist = 3)
motif consensus string: ATTAAC (consensus_score = 27)
motif positions/string s=(s1..st):
  222(ATTAAC), 47(AAAAAC), 56(ATTAAC), 12(ATTAAC), 29(AATAAC)

median string: ATTCTG (tot_dist = 3)
motif consensus string: ATTCTG (consensus_score = 27)
motif positions/string s=(s1..st):
  31(ATGCTG), 210(ATTCTG), 63(CTTCTG), 46(ATTCTG), 13(ATTTTG)