Greedy Motif Search with Pseudocounts

Input: Integers *k* and *t*, followed by a collection of strings *Dna*.

Output: A collection of strings *BestMotifs* resulting from applying *GreedyMotifSearch*(*Dna*,*k*,*t*) with pseudocounts. If at any step you find more than one *Profile*-most probable *k*-mer in a given string, use the one occurring first.

SAMPLE DATASET:

Input:

3 5
GGCGTTCAGGCA
AAGAATCAGTCA
CAAGGAGTTCGC
CACGTCAATCAC
CAATAATATTCG

Output:

TTC
ATC
TTC
ATC
TTC

The sample dataset is not actually run on your code. We check to make sure that you are actually running "Greedy Motif Search with Pseudocounts" as opposed to "Greedy Motif Search" from before.

TEST DATASET 1:

Input:

5 8

AGGCGGCACATCATTATCGATAACGATTCGCCGCATTGCC
ATCCGTCATCGAATAACTGACACCTGCTCTGGCACCGCTC
AAGCGTCGGCGGTATAGCCAGATAGTGCCAATAATTTCCT
AGTCGGTGGTGAAGTGTGGGTTATGGGGAAAGGCAGACTG
AACCGGACGGCAACTACGGTTACAACGCAGCAAGAATATT
AGGCGTCTGTTGTTGCTAACACCGTTAAGCGACGGCAACT
AAGCGGCCAACGTAGGCGCGGCTTGGCATCTCGGTGTGTG
AATTGAAAGGCGCATCTTACTCTTTTCGCTTTCAAAAAAA

Output:

AGGCG
ATCCG
AAGCG
AGTCG
AACCG
AGGCG
AGGCG
AGGCG

This dataset checks if your code has an off-by-one error at the beginning of each sequence of Dna. Notice that the all of the motifs of the solution except for the last one occur at the beginning of their respective sequences in Dna, so if your code did not check the first k-mer in each sequence of Dna, it would not find these sequences.

TEST DATASET 2:

Input:

5 8
GCACATCATTAAACGATTCGCCGCATTGCCTCGATAGGCG
TCATAACTGACACCTGCTCTGGCACCGCTCATCCGTCGAA
AAGCGGGTATAGCCAGATAGTGCCAATAATTTCCTTCGGC
AGTCGGTGGTGAAGTGTGGGTTATGGGGAAAGGCAGACTG
AACCGGACGGCAACTACGGTTACAACGCAGCAAGAATATT
AGGCGTCTGTTGTTGCTAACACCGTTAAGCGACGGCAACT
AAGCTTCCAACATCGTCTTGGCATCTCGGTGTGTGAGGCG
AATTGAACATCTTACTCTTTTCGCTTTCAAAAAAAAGGCG

Output:

AGGCG
TGGCA
AAGCG
AGGCA
CGGCA
AGGCG
AGGCG
AGGCG

This dataset checks if your code has an off-by-one error at the end of each sequence of Dna. Notice that some of the motifs of the solution occur at the end of their respective sequences in Dna, so if your code did not check the end k-mer in each sequence of Dna, it would not find these sequences.

TEST DATASET 3:

Input:

5 8

GCACATCATTATCGATAACGATTCATTGCCAGGCGGCCGC
TCATCGAATAACTGACACCTGCTCTGGCTCATCCGACCGC
TCGGCGGTATAGCCAGATAGTGCCAATAATTTCCTAAGCG
GTGGTGAAGTGTGGGTTATGGGGAAAGGCAGACTGAGTCG
GACGGCAACTACGGTTACAACGCAGCAAGAATATTAACCG
TCTGTTGTTGCTAACACCGTTAAGCGACGGCAACTAGGCG
GCCAACGTAGGCGCGGCTTGGCATCTCGGTGTGTGAAGCG
AAAGGCGCATCTTACTCTTTTCGCTTTCAAAAAAAAATTG

Output:

GGCGG
GGCTC
GGCGG
GGCAG
GACGG
GACGG
GGCGC
GGCGC

This test dataset checks if your code is correctly breaking ties when calling Profile-most Probable k-mer. Specifically, it makes sure that, when you call Profile-most Probable k-mer, in the event of a tie, you choose the first-occurring k-mer.