

Bioinformatics computing
CSE40532/60532
Homework #1

Reading:

1. Read Chapter 1 of the text (week of 8/26)
2. Read sections 3.1-3.4; 3.6 of the text (week of 9/2)
3. Read section 3.8 and handouts (week of 9/9)

Problems: (due 9/9)

1. Download the complete genomic sequence of bacteriophage lambda (accession NC_001416) and place it in your dropbox named "lambda.fasta" (2 points)
2. Write a small program called "revcomp" to reverse complement the lambda genome. Save the new sequence with a FASTA header of ">reversed" and name the resulting file as "lambda.rev.fasta". Place this file and the source code in your dropbox, and tell the TA how to compile and run in an accompanying write up also in the dropbox. (4 points)
3. Write a small program that reports nucleotide frequencies and dinucleotide frequencies of lambda. Either add the output as a table in your write up or place the code in your dropbox and tell the TA how to run it. (4 points)
4. Download the human mitochondrial genome (NC_001807). Place it in your dropbox as "human_mito.fasta." Download the Neanderthal hypervariable region (AF254446). Save it in your dropbox as "neander_sample.fasta." (2 points)
5. Write small programs to compute the log of the probability of the Neanderthal sequence under a multinomial model and a markov model of order 1 trained from the human mitochondrial sequence. Place the code in your dropbox and report the results in your write up. (5 points)
6. Generate a random sequence of 20,000 characters using a Markov model of order 3 trained on the human mitochondrial genome. Place the code for this in your dropbox and tell the TA how to run the program in your write up. (8 points)
7. Download the complete genome of *Geobacter sulfurreducens PCA* (NC_002939) and write a small program to determine the 20 most frequent 8-mers in this genome. Report the list of 20 in your write up and place the code in your dropbox with instructions (no genome please). (5 points)

HINT: DNA sequences can be represented as integers by mapping A to 00, C to 01, G to 10 and T to 11. As such AAC would be 1 (00000001). Bitwise operators can also be used for efficiency.