

Problem set IV: Simple Perl Programs

Do any one exercise. Data files are on the website.

1. Write a program to read a file (NM_021964fragment.pep) and then print its lines and elements in reverse order, the last line first.
2. Write a program to calculate the reverse complement of a strand of DNA (filename DNA3) without using the `s///` or the `tr` functions; instead, use the `substr` function to examine each base one at a time to build the complement.
3. Write a program in Perl to report on the percentage of hydrophobic amino acids in the protein sequence NM_021964fragment.pep. (You'll need to consult a molecular biology text or on-line proteomics resource to find which amino acids are hydrophobic.)
4. Write a program to determine whether sequence DNA1 or DNA2 is more GC-rich.
5. Using class exercise 10 (ex10) as a model, write a program to find motifs using regular expressions and to print out the motif sequence that was found. Use the protein sequence NM_021964fragment.pep (For example, if you enter `EE.*EE`, your program might print out a found sequence like `EETVKNDEE`.)
6. Write a Perl program to switch two nucleotide residues within a specified DNA string (DNA3) at two specified locations (nucleotide numbers).
7. Write a program to translate the DNA1 sequence to amino acids for three reading frames, starting with the first nucleotide (reading frame 0), then skipping the first one or two nucleotides to produce reading frames 1 and 2, and ignoring the last one or two nucleotides as necessary.
8. Write a program to report on the following statistics regarding the three DNA sequences DNA1, DNA2, and DNA3: their length, their GC content, and the presence or absence of poly-T sequences at the 5' (left) end of each sequence.
9. Write a program to report on the percentage of each nucleotide in the DNA sequence DNA3. Use the math operators as necessary (`+`, `-`, `*`, `/`, the `int` integer function). Write this in subroutine form, passing arguments for the name of the DNA sequence and the nucleotide you want to report.
10. Write a program in which a subroutine is called to check to see if the filename entered by the user is valid and is non-zero; repeat up to five times if invalid entries are made, then quit with an appropriate statement (curse?).
11. Write a program calling a subroutine to determine if a specified file X1, X2 or X3 is nucleotide or amino acid sequence data or is neither (i.e. invalid).
12. Write a program calling a subroutine that alters an array of lines in a file. Pass to the subroutine a reference to the array, a regular expression, and a string to replace the regular expression. All the lines of the original array should be altered by substituting the matches found for the regular expression using the replacement string. You create the original array for this exercise.
13. Write a program calling a subroutine to generate a random sequence of nucleotides. Pass the following arguments: name to assign to the DNA sequence generated; desired length of the sequence. Print out the created sequence.
14. Write a program to randomly shuffle 10 arrays of sequences of 12 nucleotides each. First print out each of the original arrays, then the shuffled array (same elements but in different order). You create the array for this exercise.
15. Write a program to randomly shuffle 12 nucleotides of an array. First print out each of the original array, then the shuffled array (same elements but in different order). You create the array for this exercise.