Episode 11 Homework

Download the data files for this episode. There are six files consisting of a single string of the DNA sequence for the APC (adenomatous polyposis coli) gene for several different species. This gene is implicated in certain forms of hereditary colon cancer when it is mutated. The species is part of the file name.

1. For this exercise you may either write a program or work at the interpreter console. Read one of the files. To read a file consisting of a single line you can use either f.read() or f.readline(), but for both you will need to strip the end-of-line marker, e.g. f.read().strip("\n\r").
    a. After reading the file, create a set with the string. Print the set. What has happened to the lengthy sequences of ATCG?
    b. Create a set consisting of the four RNA bases A, U, G, and C. Use Python set built-ins (find the documentation online if you need to look them up) to print the following:
        i. Whether RNA is a subset of DNA
        ii. The union (combined elements) of RNA and DNA
        iii. The intersection (common elements) of RNA and DNA
        iv. The difference between RNA and DNA (elements in RNA but not in DNA)
        v. The symmetric difference between RNA and DNA (elements in either but not both)

2. Write a program that reads one of the files and constructs a dictionary with the nucleotide symbol the key and the value the number of times it appears in the DNA string. Use a function to construct it and return the completed dictionary. Write another function to go through the dictionary to compute and print the proportions of each nucleotide in the gene.

3. Write a program that creates a dictionary in which the key is the species binomial name and the corresponding value is another dictionary like the one from Problem 2 above, with the key the letter and the value the number of times it appears in the sequence. This is a nested dictionary. Using this dictionary, print a comparison table of all the species for the count of A, T, C, and G in the homologous genes. Format your output table neatly.

    Hints: To extract a substring up to a particular character, use split, which returns a list, and take the appropriate element of the list. For example, to take the first field in a semicolon-separated string use s=mystring.split(";")[0]

    You may find it useful to use the format method (standard in Python 3, available in Python 2.7). To print a string with a fixed number of characters, with any extras padded with spaces, use a <W, where W is the width. For example "{0:<20}".format(mystring) to print left-justified in a width of 20.