Lab Worksheet 10 Solutions

In bioinformatics, we are often interested in determining whether or not two DNA or amino acid sequences are similar. One simple measure of similarity is called pairwise sequence identity. To calculate pairwise sequence identity, we take two sequences, count the number of positions in which both sequences share the same nucleotide or amino acid, and then divide by the total number of positions. For example, say we have these two DNA sequences:

```
       Position: 1 2 3 4 5 6
     Sequence 1: A T C G T A
     Sequence 2: A T G A G A
 Identical(y/n): y y n n n y
```

There are 3 positions that match out of 6 total positions, so the sequence identity is 50% (3/6).

Problem 1: Write code that outputs the example above to a CSV file. The CSV file should be written in **tidy** format, and it should contain one column for `position`, one column for `Sequence 1`, one column for `Sequence 2`, and one column for `Identical`. Use `sequence1` and `sequence2` to write a CSV file. Your code should generate a number for a position and check each position for a match. Once you produced a file, verify that the file was written correctly by opening it in R Studio.

In [1]:

```
sequence1 = "ATCGTA"
sequence2 = "ATGAGA"

# open a file for writing
with open("sequence_identity.csv", "w") as file:

    # write variable names as the header
    file.write("position,sequence_1,sequence_2,identical\n")

    # loop over the positions
    for i in range(len(sequence1)):

        # check a position for a match
        if sequence1[i] == sequence2[i]:
            identical = "y"
        else:
            identical = "n"

        # new line with values separated by ","
        line = str(i+1) + "," + sequence1[i] + "," + sequence2[i] + "," + ident
ical + "\n"

        # write the line to the file
        file.write(line)
```

Problem 2: Write a function that calculates the pairwise sequence identity for any two sequences of the same length. (Do not worry about properly aligning the two sequences. Sequence alignment is a concept we will return to later.) Your function should take two arguments: `seq1` and `seq2`. Make sure that your function checks for equal sequence lengths. If the input sequences are of different lengths, your function should return an error message. Otherwise, your function should return the pairwise sequence identity as a percentage.

Finally, use your function to calculate the pairwise sequence identity of the two amino acid sequences below.

```
In [2]: mouse_histone = "MARTKQTARKSTGGKAPRKQLATKAARKSAPATGGVKKPHRYRPLTVALREIRRYQKSTELL
        IRKLPFQRLVREIAQDFKTDLRFQSSAVMALQEACEAYLVGLFEDTNLCAIHAKRVTIMPKDIQLARRIRGERA"
        human_histone = "MARTKQTARKSTGGKAPRKQLATKAQRKSARATGGVKKPHRYRPGTVALREIRRYQKSTELL
        IRKLPFQRLVTEIAQDFKTDLRFQSSAVNALQEACEAYLVGLFEDTNLCAIHAKRVTIMPKDIQLARRIRGERA"

        def calc_seq_identity(seq1, seq2):

            # Check that input sequences are the same length
            if len(seq1) != len(seq2):
                return "Error, input sequences are of different lengths."

            # Start a counter for the number of matches that we find
            matches = 0

            # Loop over the sequences
            for i in range(len(seq1)):
                # At each position i, check if letters are identical
                if seq1[i] == seq2[i]:
                    matches += 1

            # Calculate a percentage
            return (matches/len(seq1))*100

        calc_seq_identity(mouse_histone, human_histone)
```

```
Out[2]: 96.32352941176471
```

Problem 3: Write a function that counts the occurence of each amino acid in a sequence and writes them to a CSV file. Amino acids in the output file should appear in the alphabetical order. Your function should write a file in **tidy** format. Use your function to output amino acid counts in sequences `mouse_histone` and `human_histone` from the previous problem. **HINT:** You can use `sorted()` to sort dictionary keys.

In [3]:

```
mouse_histone = "MARTKQTARKSTGGKAPRKQLATKAARKSAPATGGVKKPHRYRPLTVALREIRRYQKSTELL
IRKLPFQRLVREIAQDFKTDLRFQSSAVMALQEACEAYLVGLFEDTNLCAIHAKRVTIMPKDIQLARRIRGERA"
human_histone = "MARTKQTARKSTGGKAPRKQLATKAQRKSARATGGVKKPHRYRPGTVALREIRRYQKSTELL
IRKLPFQRLVTEIAQDFKTDLRFQSSAVNALQEACEAYLVGLFEDTNLCAIHAKRVTIMPKDIQLARRIRGERA"

# define function to count amino acids and write them to a file
def count_aa(seq, csv_file):

    aa_dict = {} # dictionary to store amino acid counts

    # loop over amino acids in the sequence
    for aa in seq:

        # check if amino acid is in the dictionary
        if aa in aa_dict:
            aa_dict[aa] += 1 # increment the count of an amino acid by 1
        else:
            aa_dict[aa] = 1 # set the count of an amino acid to 1

    # open a file for writing
    with open(csv_file, "w") as file:

        # sort amino acids
        aa_list=sorted(aa_dict.keys())

        # write variable names as the header
        file.write("amino_acid,count\n")

        # loop over sorted keys
        for aa in aa_list:
            # new line with values separated by ","
            line = str(aa) + ',' + str(aa_dict[aa]) + '\n'

            # write to the file
            file.write(line)

# run the function on sequences mouse_histone and human_histone
count_aa(mouse_histone,"mouse_histone_aa_count.csv")
count_aa(human_histone,"human_histone_aa_count.csv")
```