week1_solution

July 7, 2020

1 Week 1 - A solution to the assignment

1. Create a new Jupyter Notebook with one Markdown cell and one Code cell and name it as you wish e.g. `week1_exercise1`
   - Print out the names of the amino acids that would be produced by the DNA sequence "GTT GCA CCA CAA CCG" - see the DNA codon table here. Note: split the string into the individual codons and then create and use a dictionary to map between codon sequences and the amino acids they encode
   - Print each codon and its corresponding amino acid
   - Why couldn't we build a dictionary where the keys are names of amino acids and the values are the DNA codons?
   - Download the python file associated with the notebook you have created

[ ]:
```
# DNA sequence given
codon_string = "GTT GCA CCA CAA CCG"

# Split this string into the individual codons
codon_list = codon_string.split()
print(codon_list)

# Dictionary to map between codon sequences and amino acids they encode
codon_to_aminoacid = {
    "GTT": "Val",
    "GCA": "Ala",
    "CCA": "Pro",
    "CAA": "Gln",
    "CCG": "Pro"
}

print(codon_to_aminoacid)
```

[ ]:
```
# Print each codon and its corresponding amino acid
for codon in codon_list:
    print(codon, "codes for", codon_to_aminoacid[codon])
```

[ ]:
```
# Why couldn't we build a dictionary where the keys are names of amino acids␣
 ↪and the values are the DNA codons?
aminoacid_to_codon = {
```

```
    "Phe": "TTT",
    "Phe": "TTC",
    "Leu": "TTA",
    "Leu": "TTG"
}

print(aminoacid_to_codon)
```

[ ]:
```
# Why couldn't we build a dictionary where the keys are names of amino acids␣
↪and the values are the DNA codons?
aminoacid_to_codon_2 = {
    "Phe": ["TTT","TTC"],
    "Leu": ["TTA", "TTG"]
}

print(aminoacid_to_codon_2)
```

2. You are going to look at the METABRIC data file `metabric_clinical_and_expression_data.csv` on breast cancer referred above
   - Write a script that reads the file and counts how many unique patients we have data available
   - How many patients were older than 75 when diagnosed with breast cancer?
   - What were the earliest and oldest ages of diagnosis?
   - Count how many patients were treated with Chemotherapy and Radiotherapy
   - Count how many patients had less than three mutations in the genes investigated

[ ]:
```
# Write a script that reads the file and counts how many unique patients we␣
↪have data available
unique_patients = set()

with open("../data/metabric_clinical_and_expression_data.csv") as f:
    next(f)
    for line in f:
        fields = line.split(",")
        patient_id = fields[0]
        unique_patients.add(patient_id)

print("The number of unique patients is", len(unique_patients))
```

[ ]:
```
# How many patients were older than 75 when diagnosed with breast cancer?
unique_patients_older75 = set()

with open("../data/metabric_clinical_and_expression_data.csv") as f:
    next(f)
    for line in f:
        fields = line.split(",")
        patient_id = fields[0]
```

```
        age_diagnosis = float(fields[2])
        if age_diagnosis > 75:
            unique_patients_older75.add(patient_id)

print("The number of unique patients older than 75 diagnosed is",␣
 ↪len(unique_patients_older75))
```

[ ]:
```
# What were the earliest and oldest ages of diagnosis?
age_diagnosis_set = set()

with open("../data/metabric_clinical_and_expression_data.csv") as f:
    next(f)
    for line in f:
        fields = line.split(",")
        age_diagnosis = float(fields[2])
        age_diagnosis_set.add(age_diagnosis)

print("The earliest age of diagnosis is", min(age_diagnosis_set))
print("The latest age of diagnosis is", max(age_diagnosis_set))
```

[ ]:
```
# Count how many patients were treated with Chemotherapy and Radiotherapy
# Count how many patients had less than three mutations in the genes␣
 ↪investigated

unique_patients_chem_radio = set()
unique_patients_less3mut = set()

with open("../data/metabric_clinical_and_expression_data.csv") as f:
    next(f)
    for line in f:
        fields = line.split(",")
        patient_id = fields[0]
        chem = fields[6]
        radio = fields[7]
        if chem == "YES" and radio == "YES":
            unique_patients_chem_radio.add(patient_id)
        mutation_count = fields[23]
        if mutation_count != "NA":
            if int(mutation_count) < 3:
                unique_patients_less3mut.add(patient_id)

print("The number of unique patients treated with Chemotherapy and Radiotherapy␣
 ↪is", len(unique_patients_chem_radio))
print("The number of unique patients with less than 3 mutations in the genes␣
 ↪investigated is", len(unique_patients_less3mut))
```

Extra bonus: combine all of the code chunks of exercise 2 into a single chunk

3. Bonus exercise
    - Starting with an empty dictionary, count the abundance of different residue types present in the 1-letter lysozyme protein sequence and print the results to the screen in alphabetical key order.
    - Write the results to an output file

[ ]:
```
# I first downloaded the fasta file from the url manually and saved it in my␣
↪directory ../data/

residues = {}

with open("../data/B2R4C5.fasta") as f:
    next(f)
    for line in f:
        aas = list(line.strip())
        for aa in aas:
            if aa in residues:
                residues[aa] += 1
            else:
                residues[aa] = 1

for aa in sorted(residues):
    print(aa, "is present", residues[aa], "times")
```

[ ]:
```
# Write the results to an output file

with open('../data/B2R4C5.txt', 'w') as f:
    for aa in sorted(residues):
        f.write("{amino:s}\t{freq:d}\n".format(amino=aa, freq=residues[aa]))
        #f.write("%s\t%i\n" % (aa, residues[aa]))
```