The Goal for this notebook is

- ***Q1.*** To do phylogenetic tree analysis of the Nucleotide sequences provided.
- ***Q2.*** To construct a phylogenetic relationship tree diagram for the protien seqeunces provided.

THOUGHTS AND NOTES: UPGMA ASSIGNMENT

1. What even is UPGMA?

UPGMA - stands for **U**nweighted **P**air- **G**roup **M**ethod with **A**rithmetic mean.

Thats a mouthful after unpacking, so lets make it simpler to grasp -

- ***Unweighted*** - All pairwise distances contribute equally. ~~(There is an assumption here that needs to be adressed in any further attempts to this problem)~~
- ***Pair- Group*** - groups are combined in pairs (dichotomies only).
- ***Arithmetic Mean*** - pairwise distances to each group (clade) are mean distances to all members of that group.

a. *What was the assumption?*

UPGMA is a distance method and therefore needs a distance matrix. UPGMA is "ultrametric", meaning that all the terminal nodes (i.e. the sequences/taxa) are **equally distanced from the root**. In molecular terms, this means that UPGMA assumes a **molecular clock**, i.e. all lineages are evolving at a **constant rate**. In practical terms, this means that you can construct a **distance scale bar** and all the terminal nodes will be **level at position 0**.

0, representing the present.

2. How to do upgma ?

Each round of UPGMA follows the same pattern. 1. Identify the shortest pairwise distance in the matrix. This identifies the two sequences to be clustered. 2. Join the two sequences identified. 3. The pair should be linked at a depth that is half of the shortest pairwise distance. 4. The tip-to-tip distance between the joined elements will equal the shortest distance.

3. Why UPGMA?

UPGMA is a method to tackle the problem of finding out which organism is most related to which one, given we have the genetic sequences of **all** said organisms.

TLDR: Get genetic sequences as *input*, send Phylogenetic tree as *Output*.

Great. now we have a rough idea of what exactly the beast we are tackling is. At least I do. I have referred to an example provided at http://www.nmsr.org/upgma.htm to help write this program better.

4. Break it down - Divide and Conquer

THIS IS A COMPLEX PROBLEM WITH LOTS OF MOVING PARTS.

So Lets think of some simple steps to tackle this problem in.

1. Get genetic sequences as input (Needs to be tackled seperately for q1 and q2.)
2. Align These sequences, and find the distance between each pair. (We haven't *exactly* been taught this, but we have been given a link to an online tool to do this) ~~(This problem is known to be ***N P*** **HARD**)~~ I used biopython for this.
3. Create a distance matrix out of these. (should be easy enough, and simple to do for Q1 and Q2) easily done using csv
4. Do the UPGMA algo on this.

- this is gonna be involved, but should be same for both Q1 and Q2.
- yup this was involved, had to read a lot of blogs to do this. This newick form one was another pain to to. ho gaya bas ...
    5. Note the outputs and Construct a phylogenetic tree from these, and output in a given format.
- this is gonna be involved, but should be same for both Q1 and Q2.
- Used Newick form, sabse aasan tha, I swear 100% copy maara hai.
- skipping the visualisation of newick as a visual tree, there are visualisers that take newick form and show a tree, please use one , I am not doing this, it is a data visualisation technique (newick to visual tree) that is not relevant to the UPGMA method. (already exists online)

We need 4 Script files, q1a.py, q1b.py, q2a.py, q2b.py.

Questions:

1. Construct a phylogenetic relationship for the given nucleotide sequences (Nucleotide.txt).

    1. Write a script (q1a) to generate a distance matrix csv file for the sequences present in the data file. Name the distance matrix file as 'Ndistance.txt'.

- For example,
- seq1 = 'ATGCATGCAA'
- seq2 = 'ATGCATGCTA'
- Distance (seq1, seq2) = Mismatches/total length = 1/10 = 0.1

    2. Write a script (q1b) that uses 'Ndistance.txt' and generate phylogenetic relationship between the organisms using UPGMA method.

2. Construct a phylogenetic relationship for the given protein sequences (Protein.txt).

    1. Write a script (q2a) to generate a distance matrix csv file for the sequences present in the data file. Name the distance matrix file as 'Pdistance.txt'. Use BLOSUM62 for getting score values.

    2. Write a script (q2b) that uses 'Pdistance.txt' and generate phylogenetic relationship between the organisms using UPGMA method.

Explaination and Outputs of the programs :

q1a.py

This program reads the input texts and parses using biopython FASTA reader and then aligns using biopython align localds.

Output

```
 File  ./Nucleotide.txt  is accessible to read
File  ./Ndistance.txt  is accessible to write

Beginning to Read  ./Nucleotide.txt

SequenceID = NM_205222.3
Description = Gallus gallus insulin (INS), mRNA
Sequence = ATATAAATATGGGAAAGAGAATGGGGAAATTTCTACCAGTCTTCATCTCTGAGAGCAAACTTCTCTGCATCTCTTTCTCTCTTCTCTGGGCCTCCCCCAGCTCATCATGGCTCTCTGGATC

SequenceID = NM_001130093.2
Description = Canis lupus familiaris insulin (INS), mRNA
Sequence = CACCCCGACACGGCCGGCAAACAGGTCGCCATGGCCCTCTGGATGCGCCTCCTGCCCCTGCTGGCCCTGCTGGCCCTCTGGGCGCCCGCGCCCACCCGAGCCTTCGTTAACCAGCACCTGT

SequenceID = KX951416.1
Description = Labeo rohita insulin mRNA, complete cds
Sequence = CCGATGTGCTCTGAAAGCCTGGATGCAAAAACACCTTCTCTTGTCTACCATCTCTACCATTCCTTGTCCTCTGCTGCAAGAACAGTGTGACCATGGCAGTGTGGCTCCAGGCTGGTGCTCT

SequenceID = BT006808.1
Description = Homo sapiens insulin mRNA, complete cds
Sequence = ATGGCCCTGTGGATGCGCCTCCTGCCCCTGCTGGCGCTGCTGGCCCTCTGGGGACCTGACCCAGCCGCAGCCTTTGTGAACCAACACCTGTGCGGCTCACACCTGGTGGAAGCTCTCTACC

SequenceID = NM_019129.3
Description = Rattus norvegicus insulin 1 (Ins1), mRNA
Sequence = AACCCTAAGTGACCAGCTACAATCATAGACCATCAGCAAGCAGGTCATTGTTCCAACATGGCCCTGTGGATGCGCTTCCTGCCCCTGCTGGCCCTGCTCGTCCTCTGGGAGCCCAAGCCTG

SequenceID = NM_001109772.1
Description = Sus scrofa insulin (INS), mRNA
Sequence = AGCCCTCTGGGACCAGCTGTGTTCCCAGGCCACCGGCAAGCAGGTCCTCACCCCCCGCCATGGCCCTGTGGACGCGCCTCCTGCCCCTGCTGGCCCTGCTGGCCCTCTGGGCGCCCGCCCC

SequenceID = NM_173926.2
Description = Bos taurus insulin (INS), transcript variant 1, mRNA
Sequence = AGCCCCCCGCCCTCAGGACCGGCTGCATTCGAGGCTGCCAGCAAGCAGGTCCTCGCAGCCCCGCCATGGCCCTGTGGACACGCCTGGCGCCCCTGCTGGCCCTGCTGGCGCTCTGGGCCCC

SequenceID = NM_001008996.2
Description = Pan troglodytes insulin (INS), mRNA
Sequence = AGCCCTCCAGGACAGGCTGCATCAGAAGAGGCCATCAAGCAGATCACTGTCCTTCTGCCATGGCCCTGTGGATGCGCCTCCTGCCCCTGCTGGTGCTGCTGGCCCTCTGGGGACCTGACCC

SequenceID = M57671.1
Description = Octodon degus insulin mRNA, complete cds
Sequence = GCATTCTGAGGCATTCTCTAACAGGTTCTCGACCCTCCGCCATGGCCCCGTGGATGCATCTCCTCACCGTGCTGGCCCTGCTGGCCCTCTGGGGACCCAACTCTGTTCAGGCCTATTCCAG

SequenceID = U03610.1
Description = Oryctolagus cuniculus New Zealand White insulin mRNA, complete cds
Sequence = TCATCGGCTCTGCACCATGGCCTCCCTGGCCGCGCTCCTGCCCCTGCTGGCCCTGCTGGTCCTCTGCAGACTGGATCCTGCCCAGGCCTTCGTCAACCAGCACCTGTGCGGCTCTCACCTG

10 Nucleotides scanned into our List

Our Lists are :  [('NM_205222.3', Seq('ATATAAATATGGGAAAGAGAATGGGGAAATTTCTACCAGTCTTCATCTCTGAGA...GCC')), ('NM_001130093.2', Seq('CACC
[Seq('ATATAAATATGGGAAAGAGAATGGGGAAATTTCTACCAGTCTTCATCTCTGAGA...GCC'), Seq('CACCCCGACACGGCCGGCAAACAGGTCGCCATGGCCCTCTGGATGCGCCTCCTG...
['NM_205222.3', 'NM_001130093.2', 'KX951416.1', 'BT006808.1', 'NM_019129.3', 'NM_001109772.1', 'NM_173926.2', 'NM_001008996.2', 'M57
```

```
341.0 525 0.6495238095238095
362.0 539 0.6716141001855288
272.0 525 0.518095238095238
343.0 525 0.6533333333333333
336.0 525 0.64
324.0 525 0.6171428571428571
316.0 525 0.6019047619047619
327.0 525 0.6228571428571429
313.0 525 0.5961904761904762
316.0 539 0.5862708719851577
296.0 463 0.6393088552915767
352.0 463 0.7602591792656588
373.0 463 0.8056155507559395
356.0 463 0.7688984881209503
346.0 463 0.7473002159827213
332.0 463 0.7170626349892009
360.0 463 0.7775377969762419
271.0 539 0.5027829313543599
327.0 539 0.6066790352504638
305.0 539 0.5658627087198516
312.0 539 0.5788497217068646
306.0 539 0.5677179962894249
315.0 539 0.5844155844155844
291.0 539 0.5398886827458256
283.0 463 0.6112311015118791
292.0 435 0.671264367816092
280.0 434 0.6451612903225806
327.0 416 0.7860576923076923
271.0 432 0.6273148148148148
289.0 433 0.6674364896073903
351.0 463 0.7580993520518359
347.0 463 0.7494600431965442
342.0 463 0.7386609071274298
328.0 463 0.7084233261339092
337.0 463 0.7278617710583153
372.0 435 0.8551724137931035
348.0 435 0.8
330.0 435 0.7586206896551724
340.0 435 0.7816091954022989
341.0 434 0.7857142857142857
315.0 434 0.7258064516129032
332.0 434 0.7649769585253456
312.0 432 0.7222222222222222
323.0 433 0.745958429561201
321.0 433 0.7413394919168591
[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0.6495238095238095, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0.6716141001855288, 0.5862708719851577, 0, 0, 0,
```

q1b.py

Uses OOPs in Python The Node class is basically a node of a tree for upgma ,

class Node:

```
 """
Data structure to store node of a UPGMA tree
"""


    def __init__(self, left=None, right=None, up_height=0.0, down_height=0.0):
   """
   Creating a node.
   For a single taxon, set taxon name as self.left, leave right as none.
   For an operational taxonomic unit(OTU) set left and right to child nodes.

   Parameters
```

```
    ----------
    left : default = none, taxon label
    right : default = none, taxon label
    up_height : float, default = 0.0, dist to parent node, if any
    down_height : float, default = 0.0, dist to child node, if any
    """

        def leaves(self) -> list:
    """
    Method to find the taxa under any given node, effectively equivalent to
    finding leaves of a binary tree. Only lists original taxa and not OTUs.

    Returns a list of node names, not nodes themselves.
    """

        def __len__(self) -> int:
    """
    Method to define len() of a node.

    Returns the number of original taxa under any given node.
    """

        def __repr__(self) -> str:
    """
    Method to give readable print output
    """

class UPGMA:
  def __init__(self, dist_matrix: np.ndarray, taxa: list):
      """
      Initialize an UPGMA class.
      Takes a nxn distance matrix as input. A list of n taxon id is required
      in the same order as the distance matrix row/column

      Parameters
      ----------
      dist_matrix : numpy array, distance matrix of species
      taxa : list of int or str to identify taxa
      """
    def build_tree(self, dist_matrix: np.ndarray, taxa: list) -> Node:
    """
    Method to construct a tree from a given distance matrix and taxa list.

    Parameters
    ----------
    dist_matrix : np.ndarray of pairwise distances
    taxa : list of taxa id. Elements of lists have to be unique

    Returns the root node for constructed tree.
    """
        def update_distance(
    self, dist_matrix: np.ndarray, nodes: list, taxa_to_rc: dict
) -> np.ndarray:
    """
    Method to make a new distance matrix with newer node list.

    Parameters
    ----------
    dist_matrix : np.ndarray of pairwise distances for all taxa
    nodes : list of updated nodes
    taxa_to_rc : dict for taxa -> row/col id

    Returns np.ndarray of pairwise distances for updated nodes
    """
    def tree_to_newick(t) -> str:
"""
```

```
"""
Function to convert tree to Newick, slightly modified form of the tree.py version.
Takes the root node of an UPGMA tree as input
"""

And main calls all of it.
```

Outputs:

```
 File  ./Ndistance.txt  is accessible to read
[['0', '0', '0', '0', '0', '0', '0', '0', '0', '0'], ['0.6495238095238095', '0', '0', '0', '0', '0', '0', '0', '0', '0'], ['0.671614
[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0], [0.6495238095238095, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0], [0.671614100
File  ./Ntaxa.txt  is accessible to read

['NM_205222.3', 'NM_001130093.2', 'KX951416.1', 'BT006808.1', 'NM_019129.3', 'NM_001109772.1', 'NM_173926.2', 'NM_001008996.2', 'M57
(((NM_001008996.2:0.0,NM_173926.2:0.0):0.0,(NM_001109772.1:0.0,NM_019129.3:0.0):0.0):0.12365581342023683,((U03610.1:0.0,M57671.1:0.0
```

q2a.py

This program reads the input texts and parses using biopython FASTA reader and then aligns using biopython align localds.

Outputs:

```
 File  ./Protein.txt  is accessible to read
File  ./BLOSUM62.txt  is accessible to read
File  ./Pdistance.txt  is accessible to write

Beginning to Read  ./Protein.txt

SequenceID = NP_990553.1
Description = insulin preproinsulin precursor [Gallus gallus]
Sequence = MALWIRSLPLLALLVFSGPGTSYAAANQHLCGSHLVEALYLVCGERGFFYSPKARRDVEQPLVSSPLRGEAGVLPFQQEEYEKVKRGIVEQCCHNTCSLYQLENYCN

SequenceID = NP_001123565.1
Description = insulin precursor [Canis lupus familiaris]
Sequence = MALWMRLLPLLALLALWAPAPTRAFVNQHLCGSHLVEALYLVCGERGFFYTPKARREVEDLQVRDVELAGAPGEGGLQPLALEGALQKRGIVEQCCTSICSLYQLENYCN

SequenceID = ATN38177.1
Description = insulin [Labeo rohita]
Sequence = MAVWLQAGALLFLLAVSSVNANAGAPQHLCGSHLVDALYLVCGPTGFFYNPKRDVDPLMGFLPPKSAQETEVADFAFKDHAEVIRKRGIVEQCCHKPCSIFELQNYCN

SequenceID = AAP35454.1
Description = insulin [Homo sapiens]
Sequence = MALWMRLLPLLALLALWGPDPAAAFVNQHLCGSHLVEALYLVCGERGFFYTPKTRREAEDLQVGQVELGGGPGAGSLQPLALEGSLQKRGIVEQCCTSICSLYQLENYCN

SequenceID = NP_062002.1
Description = insulin-1 preproprotein [Rattus norvegicus]
Sequence = MALWMRFLPLLALLVLWEPKPAQAFVKQHLCGPHLVEALYLVCGERGFFYTPKSRREVEDPQVPQLELGGGPEAGDLQTLALEVARQKRGIVDQCCTSICSLYQLENYCN

SequenceID = NP_001103242.1
Description = insulin precursor [Sus scrofa]
Sequence = MALWTRLLPLLALLALWAPAPAQAFVNQHLCGSHLVEALYLVCGERGFFYTPKARREAENPQAGAVELGGGLGGLQALALEGPPQKRGIVEQCCTSICSLYQLENYCN

SequenceID = NP_776351.2
Description = insulin preproprotein [Bos taurus]
Sequence = MALWTRLAPLLALLALWAPAPARAFVNQHLCGSHLVEALYLVCGERGFFYTPKARREVEGPQVGALELAGGPGAGGLEGPPQKRGIVEQCCASVCSLYQLENYCN

SequenceID = NP_001008996.1
Description = insulin preproprotein [Pan troglodytes]
Sequence = MALWMRLLPLLVLLALWGPDPASAFVNQHLCGSHLVEALYLVCGERGFFYTPKTRREAEDLQVGQVELGGGPGAGSLQPLALEGSLQKRGIVEQCCTSICSLYQLENYCN

SequenceID = AAA40590.1
Description = insulin [Octodon degus]
```

```
Sequence = MAPWMHLLTVLALLALWGPNSVQAYSSQHLCGSNLVEALYMTCGRSGFYRPHDRRELEDLQVEQAELGLEAGGLQPSALEMILQKRGIVDQCCNNICTFNQLQNYCNVP

SequenceID = AAA19033.1
Description = insulin [Oryctolagus cuniculus]
Sequence = MASLAALLPLLALLVLCRLDPAQAFVNQHLCGSHLVEALYLVCGERGFFYTPKSRREVEELQVGQAELGGGPGAGGLQPSALELALQKRGIVEQCCTSICSLYQLENYCN

10 Proteins scanned into our List

Our Lists are :  [('NP_990553.1', Seq('MALWIRSLPLLALLVFSGPGTSYAAANQHLCGSHLVEALYLVCGERGFFYSPKA...YCN')), ('NP_001123565.1', Seq('MALW
[Seq('MALWIRSLPLLALLVFSGPGTSYAAANQHLCGSHLVEALYLVCGERGFFYSPKA...YCN'), Seq('MALWMRLLPLLALLALWAPAPTRAFVNQHLCGSHLVEALYLVCGERGFFYTPKA...
['NP_990553.1', 'NP_001123565.1', 'ATN38177.1', 'AAP35454.1', 'NP_062002.1', 'NP_001103242.1', 'NP_776351.2', 'NP_001008996.1', 'AAA

350.0 110 3.1818181818181817
275.0 108 2.5462962962962963
334.0 110 3.036363636363636
342.0 110 3.109090909090909
344.0 108 3.185185185185185
337.0 107 3.149532710280374
330.0 110 3.0
254.0 109 2.330275229357798
307.0 110 2.790909090909091
242.0 110 2.2
520.0 110 4.7272727272727275
466.0 110 4.236363636363636
481.0 110 4.372727272727273
473.0 110 4.3
516.0 110 4.6909090909090905
347.0 110 3.1545454545454548
453.0 110 4.118181818181818
238.0 110 2.1636363636363636
228.0 110 2.0727272727272728
228.0 108 2.111111111111111
244.0 108 2.259259259259259
242.0 110 2.2
204.0 109 1.871559633027523
227.0 110 2.0636363636363635
483.0 110 4.390909090909091
484.0 110 4.4
463.0 110 4.209090909090909
582.0 110 5.290909090909091
359.0 110 3.2636363636363637
479.0 110 4.3545454545454545
456.0 110 4.1454545454545455
437.0 110 3.9727272727272727
480.0 110 4.363636363636363
339.0 110 3.081818181818182
442.0 110 4.0181818181818185
496.0 108 4.592592592592593
481.0 110 4.372727272727273
322.0 109 2.9541284403669725
420.0 110 3.8181818181818183
459.0 110 4.172727272727273
291.0 109 2.669724770642202
412.0 110 3.7454545454545456
356.0 110 3.2363636363636363
476.0 110 4.327272727272727
332.0 110 3.018181818181818
[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [3.1818181818181817, 0, 0, 0, 0, 0, 0, 0, 0, 0], [2.5462962962962963, 2.2, 0, 0, 0, 0, 0, 0, 0, 0],
```

q2b.py

99.9% same as the q1b.py, just uses different name input files

### Outputs:

```
 File  ./Pdistance.txt  is accessible to read
[['0', '0', '0', '0', '0', '0', '0', '0', '0', '0'], ['3.1818181818181817', '0', '0', '0', '0', '0', '0', '0', '0', '0'], ['2.546296
[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0], [3.1818181818181817, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0], [2.546296296
File  ./Ptaxa.txt  is accessible to read

['NP_990553.1', 'NP_001123565.1', 'ATN38177.1', 'AAP35454.1', 'NP_062002.1', 'NP_001103242.1', 'NP_776351.2', 'NP_001008996.1', 'AAA
(((NP_001008996.1:0.0,NP_776351.2:0.0):0.0,(NP_001103242.1:0.0,NP_062002.1:0.0):0.0):0.580231790380873,((AAA19033.1:0.0,AAA40590.1:0
```

And we are Done!!!