Practical Bioinformatics (BIOL4220) - Exam 2 Topics

Exam 2 focuses on topics covered between Lab 08A and Lab 13A. However, some topics may indirectly require knowledge of topics covered between Labs 01A and 07B. To prepare, you should know:

- How to read a Python script, how to describe its overall purpose, and how to annotate key commands comments to make the script more human-readable

- How to write and run a Python script to solve a plainly stated objective

- How to define the following Python features, and how to apply them in an integrated manner:

  variables, operators, comparisons, lists, dictionaries, string methods

- How to use the following Python code constructs:

  If-statements, for-loops, list comprehensions, functions, modules, file handlers

- How to create, modify, and use objects from the following BioPython modules:

  Seq, CodonTable, SeqIO, AlignIO

- How to use select features from NumPy

  ndrray, indexing ndarray elements, arraywise operators, access elements, etc.

- When provided a with a genetic code and a list of amino acid physicochemical properties, how to compute various summary statistics for molecular sequences:

  nucleotide composition, GC-content, phylogenetically informative sites, amino acid translations and frequencies, physicochemical properties of amino acids, codon usage bias

- When provided with a genetic code, how to compute *dN/dS* for two sequences by hand using the counting method

- How to interpret and reason with various phylogenetic estimates, including:

  branch length estimates, phylogenetic relationships, dN/dS scores

Below are questions similar to those that might appear on Exam 2:

- Write Python code to perform the following tasks:

    - Initialize a string to read “The four nucleotides are A, C, G, and T.” Apply a string method to count the number of occurrences of the letter ‘T’. Modify the method to count both uppercase and lowercase occurrences of ‘T’. Replace ‘four’ with ‘4’.
    - Define a dictionary with three items with keys ‘E’, ‘M’, and ‘N’ and values ‘envelope’, ‘membrane’, ‘nucleocapsid’. Add the item with key ‘S’ and value ‘spike’ to the dictionary.
    - Define a NumPy array that has four rows and four columns, with values 1 through 16 as elements. Assign the values [[-1, -2], [-3, -4]] to the submatrix defined by the second and third rows and second and third columns using a single command.
    - Define a list with the values [1, 2, 3, 0]. Replace the last element in the list with the value 4. Call a list comprehension that squares the value for each element in the original list.
    - Write a function that accepts an input filename, an output filename, and a string as arguments. The function will read each line in the input file, then write that line to the output file only if it does _not_ contain the string argument.

- You are given a file that is formatted as follows:

```
accession,gene,num_sites,r_score,m_score
MW342724,spike,3861,381,239
MW342724,envelope,5851,327,121
MW342724,membrane,9511,661,550
MW342724,nucleocapsid,2393,217,192
MW342616,spike,3867,383,231
MW342616,envelope,5855,323,175
MW342616,membrane,9541,631,554
MW342616,nucleocapsid,2392,241,197
...
```

  Write a Python script that reads the file to populate a dictionary, where the key for each item is the gene name, and the value for each item is itself a dictionary. Items for these internal dictionaries use the accession as the key and have a value equal to `n_sites * (r_score/m_score)`. The resulting dictionary should then

```
{ ‘spike’: { ‘MW342724’: 6154.983263598327,
             ‘MW24616’: 6244.116883116883 },
  ‘envelope’: { ‘MW342724’: 15812.20661157025,
                ‘MW24616’: 10806.657142857142 },
  ‘membrane’: { ‘MW342724’: 11430.492727272727,
                ‘MW24616’: 10867.095667870037 },
  ‘nucleocapsid’: { ‘MW342724’: 2704.5885416666665,
                    ‘MW24616’: 2926.253807106599 } }
```

- You are given a file that contains the following Python code:

```
from Bio import SeqIO
from Bio.Data import CodonTable
standard_table = CodonTable.unambiguous_dna_by_name["Standard"]
stop = standard_table.stop_codons
infn = 'all_E.fasta'
outfn = 'all_E.phy'
d = SeqIO.to_dict(SeqIO.parse(infn,'fasta'))
n_seq = len(d)
n_sites = len( d[next(iter(d))] )
s = str(n_seq) + ' ' + str(n_sites) + '\n'
for k,v in d.items():
      seq = ''
      c1 = str(v.seq[0::3])
      c2 = str(v.seq[1::3])
      c3 = str(v.seq[2::3])
      n_codons = min([len(c1), len(c2), len(c3)])
      for i in range(n_codons):
      c123 = c1[i]+c2[i]+c3[i]
      if c123 in stop:
            c123 = 'NNN'
      seq += c123
      s += k[:30] + '  ' + seq + '\n'

outf = open(outfn, 'w')
outf.write(s)
outf.close()
```

Comment each line in the code above, explaining what it does. Describe the overall purpose of the code.

- What does the dN/dS parameter represent? How do you interpret dN/dS = 1? dN/dS < 1? dN/dS > 1? What might cause dN/dS values to differ for two separate genes? What might cause dN/dS values to vary across sites within a single gene?

- You are given sequences from two protein-coding genes, each with four codons.

```
AGTCCATGTACTACC
AGTCCATTACGATAC
```

Compute the number of synonymous substitutions, nonsynonymous substitutions, synonymous sites, and nonsynonymous sites. Use these four numbers to compute the dN/dS ratio.

- You are given the sequence alignment for a protein-coding gene

  ```
  Sp1 AGTTCTAGACCGGTT
  Sp2 AGGTTTCGATCGGGT
  Sp3 AGGTCTCGAACGGTT
  Sp4 AGTTCTCGATCGGTT
  ```

  Compute the GC content for each sequence. Identify all of the phylogenetically informative sites. Translate the alignment into amino acids. What percent of amino acids are hydrophobic?

- Suppose you estimate separate molecular phylogenies for five species (A, B, C, D, E) for each of two genes (X and Y). The phylogeny for gene X is

  ```
  ((A:0.1,B:0.2):0.1,(C:0.5,(D:0.2,E:0.3):0.2):0.3);
  ```

  and the phylogeny for gene Y is

  ```
  ((A:0.02,B:0.04):0.5,(C:0.2,(D:0.1,E:0.05):0.1):0.3);
  ```

  Gene X has 500bp and Gene Y has 1000bp. Which gene is expected to have experienced a greater total number of substitutions (i.e. per gene, not per site)?

- Name two topics that you think the course should have covered, but did not. Name one topic that was covered that you think should be covered in greater depth, and one topic that the course did cover but you did not find particularly interesting or useful.