Final Exam QBIO 401

This exam should only be your own work. Do not consult with anyone else (student or non-student). You can use your notes, my lecture slides, the lecture recordings, HW solutions, and any books, papers, or web resources. If you have questions, I will hold my usual office hours during finals period (or you can email me).

Turn in a pdf of a Jupyter notebook with your code, the requested outputs, and the answers to the questions. Submit your solutions on Blackboard before midnight on Friday, December 8.

Files attached to this assignment on Blackboard: omicron.fasta, delta.fasta, lecture2functions.py, enzymelist.csv, and abbwdbc.csv.

Note this exam has six pages: this page and five pages of problems (two numbered problems).

1. (15 pts) The statement of this problem includes some background on restriction enzymes and gel electrophoresis. Be sure to answer parts (a) – (f).

Attached to this assignment are two files with coronavirus variant sequences in the FASTA format: “omicron.fasta” and “delta.fasta.” Use the loadFASTA function from “lecture2functions.py” (this file is also attached to this assignment) to get the sequence data in these files.

(a) (1 pt) How long (in base pairs) is the sequence in the “omicron.fasta” file? How long (in base pairs) is the sequence in the “delta.fasta” file?

A restriction enzyme is an enzyme that cuts DNA sequences at a specific target (also called a restriction site). Restriction enzymes are believed to have originated in bacteria as a defense mechanism against invading viruses (there is a way for the bacteria to protect its own DNA so that only the virus is cut up). Molecular biologists have used restriction enzymes to manipulate DNA in lots of applications. There are more than 3,000 known restriction enzymes.

The “enzymelist.csv” file attached to this assignment is a list of nine restriction enzymes and their targets. **Note**: in the targets the symbol “N” denotes A, C, G, or T; the symbol “Y” denotes C or T; the symbol “R” denotes A or G; and the symbol “W” denotes A or T. The symbols “N”, “Y”, “R”, and “W” will **not** appear in the FASTA files.

A restriction enzyme will cut a sequence at **every** copy of its target. Below are some short examples:

(i) The restriction enzyme AaaI (target CGGCCG) has two copies of its target in the following sequence:

ATTGGACCATTA**CGGCCG**AA**CGGCCG**TAGGACCTTTTTTTTGGTCCAAGGTCCTC

It will then cut the sequence into three pieces (of lengths 13, 8, and 34):

ATTGGACCATTA**C**   **GGCCG**AA**C**   **GGCCG**TAGGACCTTTTTTTTGGTCCAAGGTCCTC

(ii) The restriction enzyme AanI (target TTATTA) has zero copies of its target in the following sequence:

ATTGGACCATTACGGCCGAACGGCCGTAGGACCTTTTTTTTGGTCCAAGGTCCTC

So it will not cut the sequence, and the sequence will remain whole in one piece.

(iii) The restriction enzyme AfII (target GGWCC) has four copies of its target in the following sequence:

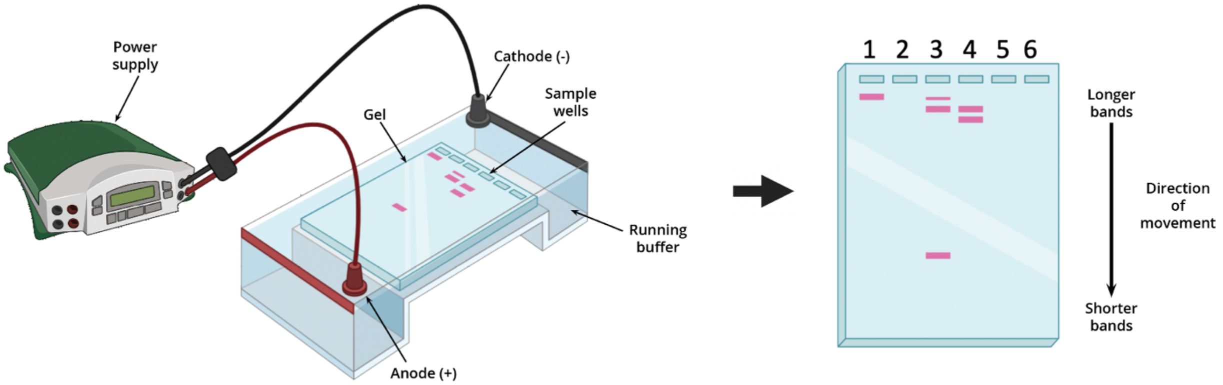
ATT**GGACC**ATTACGGCCGAACGGCCGTA**GGACC**TTTTTTTT**GGTCC**AA**GGTCC**TC

It will then cut the sequence into five pieces (of lengths 4, 25, 13, 7, and 6):

ATT**G**   **GACC**ATTACGGCCGAACGGCCGTA**G**   **GACC**TTTTTTTT**G**   **GTCC**AA**G**   **GTCC**TC

In gel electrophoresis, biomacromolecules (such as DNA, RNA, or proteins) are injected into one side of a gel, and an electric field is applied. Any charged molecules will move across the gel, propelled by the electric force. The biomacromolecules are stained so we can see how far across the gel they have travelled. The key is that larger biomacromolecules move through the gel more slowly than smaller biomacromolecules. Gel electrophoresis can be used to separate biomacromolecules. It can also be used to identify which biomacromolecules are present in a sample when compared with reference samples.

In the figure below, look at the gel on the right-hand side of the image. There are six lanes (vertical columns) where a DNA sample could be injected. The samples are stained pink. Pink bands near the top of the gel are longer DNA pieces than pink bands closer to the bottom of the gel. In lane #1 we see that this sample had one (relatively long) DNA piece. In lanes #2, #5, and #6 no samples were injected. In lane #3, we see that this sample had three DNA pieces of different lengths (because there are three distinct bands). In lane #4, we see that this sample had two DNA pieces of different lengths. We observe the longer DNA piece in lane #4 (the one closer to the top) is the same length as the middle DNA piece in lane #3 (because these two bands are at the same vertical position). We also observe the shorter DNA piece in lane #4 (the one closer to the bottom) is a different length than all of the DNA pieces in lane #3 (because there is no band in lane #3 at the same vertical position). Similarly, the longest DNA piece in lane #3 (the thin band near the top) and the shortest DNA piece in lane #3 (the band way down the bottom of the gel) are different lengths than all of the DNA pieces in lane #4.



In this problem we are going to investigate whether we can use restriction enzymes and gel electrophoresis to distinguish between the omicron and delta variants without sequencing the samples. The goal is to find a restriction enzyme that will cut the omicron and delta variants differently. And to make it easy to distinguish on the gel, we want at least one of the DNA pieces cut from the omicron variant to be many base pairs different in length (not just a few base pairs) than all of the pieces cut from the delta variant. For example in the figure above, if lanes #3 and #4 were different DNA samples (after they had been separately cut by the same restriction enzyme) we could tell that these two DNA samples had different DNA sequences (without sequencing the samples).

(b) (6 pts) Write a function (or functions) that will take as input both a sequence and one target from the “enzymelist.csv” file and will output the number of copies of that target in the sequence.

I want to emphasize that the function should take the target directly from the “enzymelist.csv” file without you making any alterations by hand. Though the file has only nine restriction enzymes, the same function should work if the enzyme list had thousands of restriction enzymes (and you wouldn’t want to be making manual alterations with such a long list).

(c) (2 pts) Separately for each target in the “enzymelist.csv” file, use the function you wrote in part (b) to compute how many copies of that target are in the omicron sequence. (You should compute nine numbers, one for each restriction enzyme target.)

I want to emphasize that throughout problem #1, we are considering restriction enzymes acting separately (we are **not** considering different restricting enzymes acting together).

(d) (2 pts) Repeat part (c) for the delta sequence.

(e) (1 pt) Based on your answers to parts (c) and (d), you should see that there is exactly one restriction enzyme such that the number of copies of the target is different in the omicron and delta sequences. Which restriction enzyme is it?

(f) (3 pts) Write a function (or functions) that will take as input both a sequence and one restriction enzyme target and will output a list of the lengths of the pieces the sequence will be cut into by the restriction enzyme. For the restriction enzyme you found in part (e), run this function separately on the omicron and delta sequences (the outputs will then be two lists of numbers).

The third column in the file “enzymelist.csv” tells the exact location of the cut within the target. This location is different for each restriction enzyme. This information will only affect the lengths of the cut pieces by a few base pairs (and it is OK if your computed lengths are off by just a few base pairs). You should have found that one of the pieces cut from the omicron variant is more than 100 base pairs longer than any of the pieces cut from the delta variant, so we can use restriction enzymes and gel electrophoresis to distinguish between omicron and delta without sequencing.

Comment:

Restriction enzymes cut double-stranded DNA, so the single-stranded RNA coronavirus will be transformed into double-stranded DNA before cutting with the restriction enzyme. Also, the sequences in the two FASTA files are in the 5’ to 3’ direction. The targets in the “enzymelist.csv” file are also in the 5’ to 3’ direction. So there is no need to do anything with reverse complements, just follow the short examples.

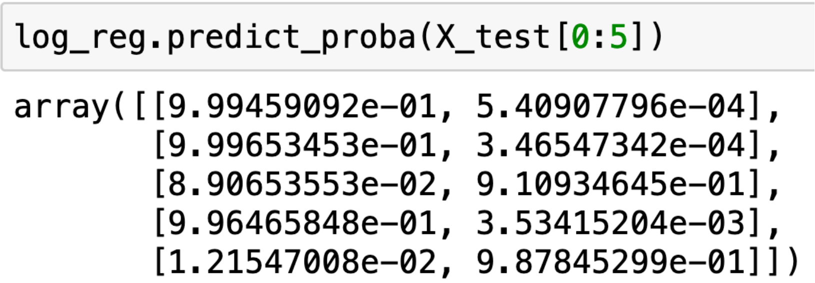
2. (15 pts) The statement of this problem includes some background on logistic regression. Be sure to answer parts (a) – (f).

   For the second problem we are going to use the cancer file “abbwdbc.csv” attached to this assignment. This file is an abbreviated version of the file from HW 8 (many of the columns from the file in HW 8 have been deleted). Similar to HW 8, we are going to use logistic regression to try to predict the diagnosis column.

   (a) (1 pt) How many rows have diagnosis “B” and how many rows have diagnosis “M”? (These numbers are relatively balanced, so there is no need to use any of the methods from the unbalanced data lecture, but it is a good idea to check.)

   Split the data into training and test sets (use test_size=.25 and random_state=123), standardize the X-variables, and fit the logistic regression model on the training set.

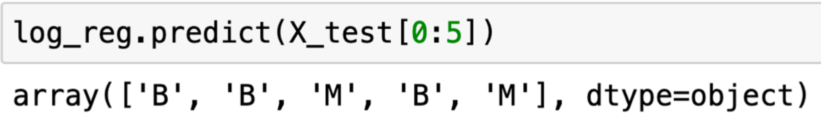
   The logistic regression model computes the probability that a diagnosis will be “B” or “M” for a given set of X-variables. In the screenshot below, the first column is the probability the diagnosis will be “B”. We will call this first column p0. (The second column is 1 – p0, and is the probability the diagnosis will be “M”.)



```
log_reg.predict_proba(X_test[0:5])

array([[9.99459092e-01, 5.40907796e-04],
       [9.99653453e-01, 3.46547342e-04],
       [8.90653553e-02, 9.10934645e-01],
       [9.96465848e-01, 3.53415204e-03],
       [1.21547008e-02, 9.87845299e-01]])
```

   The threshold for prediction is 0.5, so if p0 > 0.5 in a row the model predicts “B” for that row and if p0 < .5 in a row the model predicts “M” for that row (compare the probabilities in the screenshot above to the predictions in the screenshot below).



```
log_reg.predict(X_test[0:5])

array(['B', 'B', 'M', 'B', 'M'], dtype=object)
```

In this problem we are going to study whether the model is more likely to make the correct prediction if p0 is far from the threshold of 0.5 (so near 0 or 1) than if p0 is near 0.5.

(b) (4 pts) Plot a histogram of p0 for all the rows in the test set.

(c) (4 pts) Compute the accuracy (the percentage of rows the model prediction is correct) and the confusion matrix for all the rows in the test set.

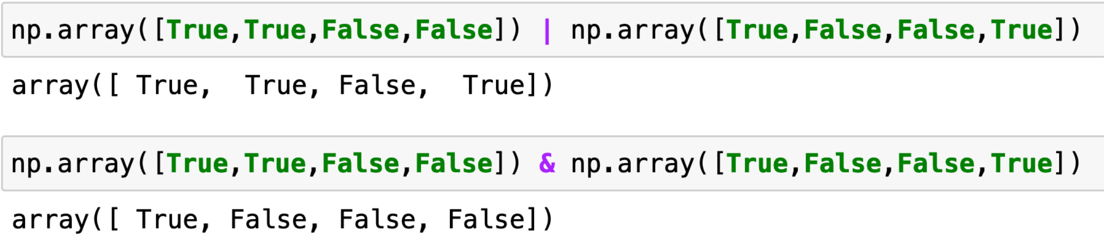
(d) (3 pts) For part (d) we will restrict ourselves to just those rows in the test set such that p0 is less than 0.2 or greater than 0.8. For just these rows, compute the accuracy and the confusion matrix.

(e) (2 pts) For part (e) we will restrict ourselves to just those rows in the test set such that p0 is between 0.2 and 0.8. For just these rows, compute the accuracy and the confusion matrix.

(f) (1 pt) Based on your results in parts (c), (d), and (e), is the model more likely to make the correct prediction if p0 is far from the threshold of 0.5 (so near 0 or 1) than if p0 is near 0.5?

Comment:

The “&” symbol is a bitwise “and” operator for arrays and the “|” symbol is a bitwise “or” operator for arrays. Examples in the screenshot below.



```
np.array([True,True,False,False]) | np.array([True,False,False,True])

array([ True,  True, False,  True])

np.array([True,True,False,False]) & np.array([True,False,False,True])

array([ True, False, False, False])
```