Biol 4150/6150 Midterm Exam 1 Fall 2021

1. HMMs (20 pts)
In a fictional organism, the median length of introns is 160 bp. The median length of exons is 200 bp. The frequency of G:C base pairs in exons is 40%, and 30% within introns.
Construct a table that shows the transition and emission probabilities for an HMM for 5' splice site prediction in this species, that incorporates an additional requirement that the 2nd base after the G in the intron is a T 99% of the time, and a C in 1% of introns.

I constructed the emission and transition probabilities with the specified length of 160 bp for introns and 200 bp for exons along with the G:C frequency of 40% in exons and 30% in introns. I have also incorporated the criterion of having a 'T' base 99% of the time and a 'C' base 1% of the time after 'G' in the intron.

The reference sequence that I utilized was:

Exon -> 5'SS -> Intron(T/C) -> Intron -> End

The transition probabilities are as follows:

	Intron	Splice Site	First base of intron (T/C)	Exon	End
Intron	0.994	0	0	0	0.006
Splice Site	0	0	1	0	0
Exon	0	0.005	0	0.995	0

The emission probabilities are as follows:

	A	G	C	T
Exon	0.3	0.2	0.2	0.3
Splice	0	1	0	0
First base of intron (T/C)	0	0	0.01	0.99
Intron	0.35	0.15	0.15	0.35

2. (20 pts) Fill in the table below for a Needleman-Wunch global alignment between the two sequences, using the BLOSUM 62 scoring matrix, and a uniform gap penalty of -2. Highlight the cells that give the optimal alignment path and write out the optimal pairwise alignment indicated from the traceback. Hint: you may find baba.sourceforge.net very helpful for this, and you can copy and paste a screenshot image from the program instead of filling out the table.

		G	E	N	E	S

A						
G						
E						
N						
T						
S						

Resulting alignment (be sure your characters appear properly aligned by using a non-proportional font):

Please find the global alignment of "GENES" vs "AGENTS" using the Needleman-Wunch dynamic programming.

I had to first install and get java running on my system upon which I downloaded the baba.sourceforge.net file on it. I changed the gap penalty to be a uniform -2 in all of the cells and the scoring matrix to be BLOSUM62. The results obtained is pasted in the screenshot above. I have highlighted the alignment in a green font.

Additionally, I also calculated the traceback and this is the best possible alignment in accordance to the traceback

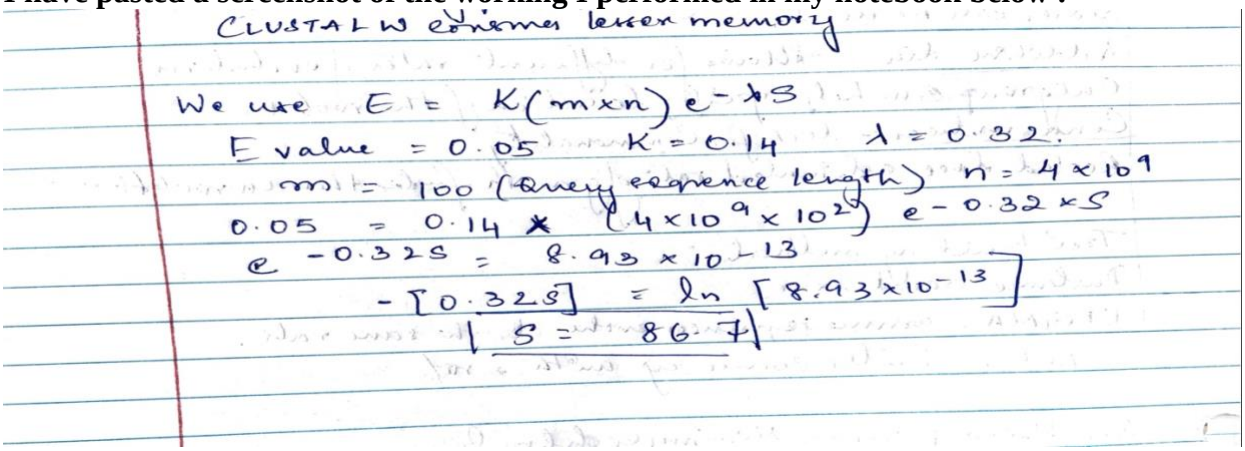
A	**G**	**E**	**N**	**T**	**S**
	G	**E**	**N**	**E**	**S**
-2	**6**	**5**	**6**	**-1**	**4**

The total is alignment score obtained as illustrated avove is +18.

3. NCBI BLAST – perform BLAST searches with the SARS-CoV-2 protein: YP_009725307

a)(5 pts) What is the minimum sequence similarity score S for a hit to be included when a BLAST search is run with threshold E value set to 0.05? Estimated lambda = 0.32, estimated K = 0.14, and the database size is 4 billion. The query sequence length is 100.

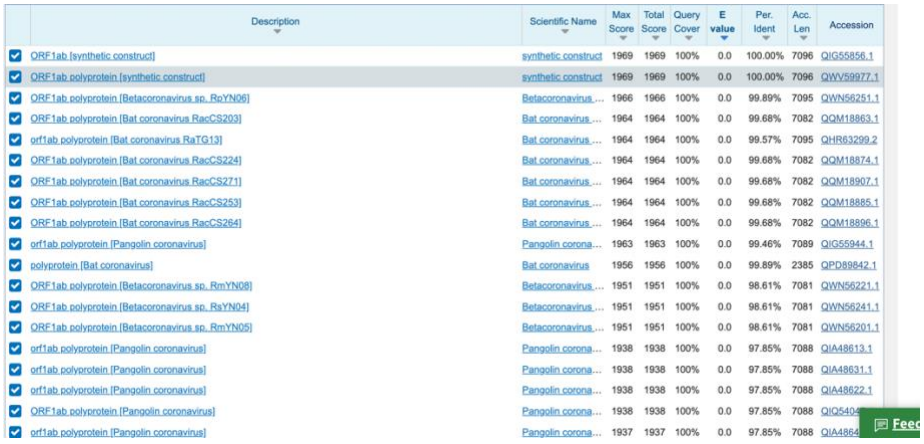
I have pasted a screenshot of the working I performed in my notebook below :



b) (5 pts) What blast algorithm should you use for a fast search of highly similar protein sequences?

The blast algorithm which can be used for searching highly similar protein sequences is 'Quick BlastP' or 'SmartBlast'

c) (5 pts) How can you exclude other SARS-CoV-2 isolates in your search results? Paste a screenshot of your search results.
SARS-coV-2 isolates can be excluded by using the exclude option (Tax id for exclusion is : 2697049)during the blast search. A screenshot of the results I obtained after exclusion is presented below:



Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
ORF1ab [synthetic construct]	synthetic construct	1969	1969	100%	0.0	100.00%	7096	QIG55856.1
ORF1ab polyprotein [synthetic construct]	synthetic construct	1969	1969	100%	0.0	100.00%	7096	QWV59977.1
ORF1ab polyprotein [Betacoronavirus sp. RpYN06]	Betacoronavirus ...	1966	1966	100%	0.0	99.89%	7095	QWN56251.1
ORF1ab polyprotein [Bat coronavirus RacCS203]	Bat coronavirus ...	1964	1964	100%	0.0	99.68%	7082	QQM18863.1
orf1ab polyprotein [Bat coronavirus RaTG13]	Bat coronavirus ...	1964	1964	100%	0.0	99.57%	7095	QHR63299.2
ORF1ab polyprotein [Bat coronavirus RacCS224]	Bat coronavirus ...	1964	1964	100%	0.0	99.68%	7082	QQM18874.1
ORF1ab polyprotein [Bat coronavirus RacCS271]	Bat coronavirus ...	1964	1964	100%	0.0	99.68%	7082	QQM18907.1
ORF1ab polyprotein [Bat coronavirus RacCS253]	Bat coronavirus ...	1964	1964	100%	0.0	99.68%	7082	QQM18885.1
ORF1ab polyprotein [Bat coronavirus RacCS264]	Bat coronavirus ...	1964	1964	100%	0.0	99.68%	7082	QQM18896.1
orf1ab polyprotein [Pangolin coronavirus]	Pangolin corona...	1963	1963	100%	0.0	99.46%	7089	QIG55944.1
polyprotein [Bat coronavirus]	Bat coronavirus	1956	1956	100%	0.0	99.89%	2385	QPD89842.1
ORF1ab polyprotein [Betacoronavirus sp. RmYN08]	Betacoronavirus ...	1951	1951	100%	0.0	98.61%	7081	QWN56221.1
ORF1ab polyprotein [Betacoronavirus sp. RaYN04]	Betacoronavirus ...	1951	1951	100%	0.0	98.61%	7081	QWN56241.1
ORF1ab polyprotein [Betacoronavirus sp. RmYN05]	Betacoronavirus ...	1951	1951	100%	0.0	98.61%	7081	QWN56201.1
orf1ab polyprotein [Pangolin coronavirus]	Pangolin corona...	1938	1938	100%	0.0	97.85%	7088	QIA48613.1
orf1ab polyprotein [Pangolin coronavirus]	Pangolin corona...	1938	1938	100%	0.0	97.85%	7088	QIA48631.1
orf1ab polyprotein [Pangolin coronavirus]	Pangolin corona...	1938	1938	100%	0.0	97.85%	7088	QIA48622.1
ORF1ab polyprotein [Pangolin coronavirus]	Pangolin corona...	1938	1938	100%	0.0	97.85%	7088	QIQ5404
orf1ab polyprotein [Pangolin coronavirus]	Pangolin corona...	1937	1937	100%	0.0	97.85%	7088	QIA4864

d) (5 pts) How can you refine your search to determine if distantly related sequences exist outside of coronaviruses (SARS-CoV-2 is a member of the betacoronavirus subfamily)? What program(s) would you use?
Paste a screenshot of your results of a search for distantly related sequences in other viruses or organisms excluding all coronaviruses.

I have excluded betacoronavirus in the screenshot below, the program I used to run it was BLAST-P.

Sequences producing significant alignments — Download — Select columns — Show 100

select all 100 sequences selected — GenPept — Graphics — Distance tree of results — Multiple alignment — MSA Viewer

Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
ORF1ab [synthetic construct]	synthetic construct	1969	1969	100%	0.0	100.00%	7096	QIG55856.1
ORF1ab polyprotein [synthetic construct]	synthetic construct	1969	1969	100%	0.0	100.00%	7096	QWV59977.1
ORF1ab polyprotein [Bat coronavirus]	Bat coronavirus	1968	1968	100%	0.0	99.89%	7094	UAY13264.1
ORF1ab polyprotein [Bat coronavirus]	Bat coronavirus	1968	1968	100%	0.0	99.89%	7095	UAY13240.1
ORF1ab polyprotein [Bat coronavirus]	Bat coronavirus	1968	1968	100%	0.0	99.89%	7088	UAY13252.1
ORF1ab polyprotein [Bat coronavirus]	Bat coronavirus	1968	1968	100%	0.0	99.89%	7088	UAY13228.1
ORF1ab polyprotein [Bat coronavirus]	Bat coronavirus	1967	1967	100%	0.0	99.89%	7087	UAY13216.1
polyprotein [Bat coronavirus]	Bat coronavirus	1956	1956	100%	0.0	99.89%	2385	QPD89842.1
polyprotein [recombinant SARS coronavirus]	recombinant SA...	1914	1914	100%	0.0	96.35%	7073	ACJ60680.1
orf1ab polyprotein [Bat coronavirus]	Bat coronavirus	1890	1890	100%	0.0	95.17%	7049	QTJ30143.1
replicase 1B [recombinant coronavirus]	recombinant cor...	1880	1880	99%	0.0	95.77%	2695	ACJ60702.1
ORF1ab polyprotein [Bat coronavirus]	Bat coronavirus	1448	1448	98%	0.0	73.45%	6877	YP_009824989.2
P1ab [Bat coronavirus]	Bat coronavirus	1434	1434	100%	0.0	72.13%	998	ANA96021.1
P1ab [Bat coronavirus]	Bat coronavirus	1433	1433	100%	0.0	72.13%	998	ANA96020.1
orf1ab polyprotein [Bat coronavirus]	Bat coronavirus	1430	1430	99%	0.0	72.15%	6865	QEH60462.1
P1ab [Bat coronavirus]	Bat coronavirus	1427	1427	99%	0.0	72.17%	991	ANA96019.1
P1ab [Bat coronavirus]	Bat coronavirus	1427	1427	100%	0.0	71.96%	991	ANA96022.1
P1ab [Bat coronavirus]	Bat coronavirus	1425	1425	100%	0.0	71.96%	991	ANA96023.1

In the second trial I excluded the entire family of coronaviruses by using exclude option on coronaviridae family (Tax ID : 11118) my results were as the screenshot presented below, again I used the BLAST-P program.

Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
ORF1ab [synthetic construct]	synthetic construct	1969	1969	100%	0.0	100.00%	7096	QIG55856.1
ORF1ab polyprotein [synthetic construct]	synthetic construct	1969	1969	100%	0.0	100.00%	7096	QWV59977.1
polyprotein [recombinant SARS coronavirus]	recombinant SA...	1914	1914	100%	0.0	96.35%	7073	ACJ60680.1
replicase 1B [recombinant coronavirus]	recombinant cor...	1880	1880	99%	0.0	95.77%	2695	ACJ60702.1
ORF1ab polyprotein [synthetic construct]	synthetic construct	1346	1346	68%	0.0	100.00%	1129	QUS47417.1
ORF1ab polyprotein [synthetic construct]	synthetic construct	1306	1306	99%	0.0	66.59%	7178	AAX23975.1
1ab polyprotein [synthetic construct]	synthetic construct	1184	1184	99%	0.0	62.07%	6631	AMK49142.1
ORF1B [recombinant coronavirus]	recombinant cor...	1142	1142	99%	0.0	59.03%	2678	ACJ60705.1
ORF1ab polyprotein [synthetic construct]	synthetic construct	828	828	41%	0.0	100.00%	981	QUS47415.1
ORF1b polyprotein [Pacific salmon nidovirus]	Pacific salmon n...	625	625	89%	0.0	42.67%	2717	QEG08237.1
RNA dependent RNA polymerase family protein [Escherichia coli 1-176-05_S1_C1]	Escherichia coli ...	386	386	25%	3e-124	76.25%	241	EYD92547.1
hypothetical protein [Escherichia coli]	Escherichia coli	353	353	23%	1e-111	76.71%	221	WP_205957693.1
replicase poly1ab domain protein [Escherichia coli 1-110-08_S3_C1]	Escherichia coli ...	162	162	9%	7e-43	77.17%	94	EYE15391.1
polyprotein 1ab [Bellinger River virus]	Bellinger River v...	161	161	82%	7e-36	26.13%	7459	YP_009755843.1
pp1b [Python nidovirus]	Python nidovirus	160	160	82%	1e-35	24.78%	2341	AII00826.1
pp1ab [Ball python nidovirus 1]	Ball python nido...	159	159	82%	2e-35	25.56%	8108	YP_009052475.1
ORF1B [Serpentovirinae sp.]	Serpentovirinae ...	159	159	82%	3e-35	25.44%	2237	QFU19804.1
pp1ab replicase polyprotein [Ball python nidovirus 1]	Ball python nido...	157	157	82%	9e-35	25.44%	8170	AUS29609.1
ORF1AB [Serpentovirinae sp.]	Serpentovirinae ...	157	157	82%	1e-34	25.44%	5439	QFU19796.1
ORF1B [Serpentovirinae sp.]	Serpentovirinae ...	155	155	82%	3e-34	24.94%	2238	QFU19734.1
ORF1AB [Serpentovirinae sp.]	Serpentovirinae ...	155	155	82%	6e-34	25.28%	7328	QFU19726.1
ORF1B [Serpentovirinae sp.]	Serpentovirinae ...	153	153	82%	2e-33	25.78%	2238	QFU19714.1
ORF1b [Morelia viridis nidovirus]	Morelia viridis ni...	152	152	82%	4e-33	24.94%	2322	YP_009408170.1
ORF1AB [Serpentovirinae sp.]	Serpentovirinae ...	152	152	82%	4e-33	26.39%	7876	QFU19706.1

4. Perform a multiple sequence alignment of 20 different, diverse myoglobin protein sequences, from different mammals. Add one or two outgroup myoglobin sequences. Be sure to use full-length or Refseq sequences. Use the MUSCLE program included in the MEGA package.

a) (5 pts) What gap opening and extension penalties did you use? Which distance method did you use for the 1st 2 iterations and for additional iterations?

The gap opening penalties being used is -2.90 and the gap extension penalty is 0
I used CLUSTER method(UPGMA)for iterations 1,2 and the CLUSTER(UPGMA) method for the other iterations as well

b) (5 pts) upload your multiple sequence alignment as a pdf document.

I have uploaded the pdf – Please note that I choose sequences with similar lengths and the gaps obtained are considerably less but present in the pdf if looked upon carefully!

c) (10 pts) list some key differences between MUSCLE and ClustalW in how they perform multiple sequence alignments. How does MUSCLE deal with the problem of greediness of progressive alignments?

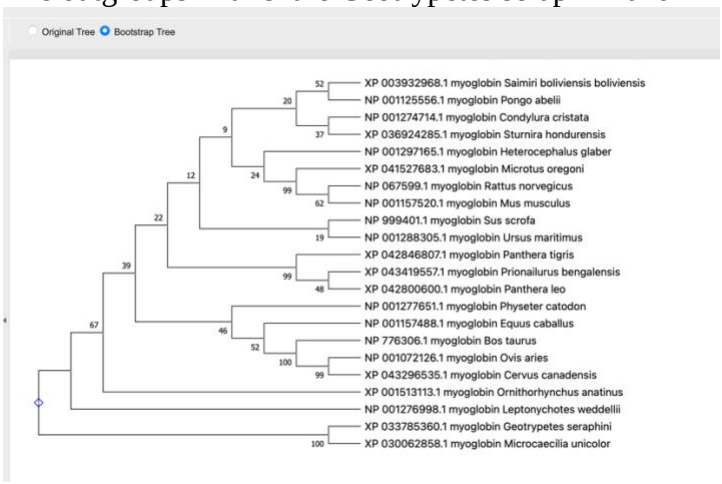
The key differences between MUSCLE and ClustalW in terms of how they perform multiple sequence alignment are:

ClustalW	Muscle
• **Comparatively lesser alignment accuracy** • **Clustal W consumes lesser memory for the alignment process** • **Suitable for short alignments of about 50 sequences** • **Suitable for sequences with low xhomology N-terminal or C-terminal extensions** • **Uses a progressive algorithm and hence mistakes that maybe produced initially are seldom corrected** • **Constructs guide tree using neighbor-joining method** • **Similarity measures used: pairwise alignment is done through the k-tuple method.** • **Even though more than one optimum pairwise alignment is possible ClustalW decides the optimum pairwise alignment at the outset**	• **Alignment Accuracy is better (in terms of sum of pairs/Total column scores)** • **Muscle consumes more computer memory** • **Suitable for very large datasets of over 1000 sequences** • **Not suitable for sequences with low homology N-terminal and C-terminal extensions.** • **Muscle uses an iterative algorithm and allows re-optimizations of columns during the whole process** • **Constructs guide tree typically using UPGMA method** • **Similarity measures used: Fractional D obtained through global alignment of two sequences and measures obtained through k-mer counting followed by kimura distance method where the initial tree is re-estimated** • **More than one optimum pairwise alignment is possible, and muscle takes this into consideration**

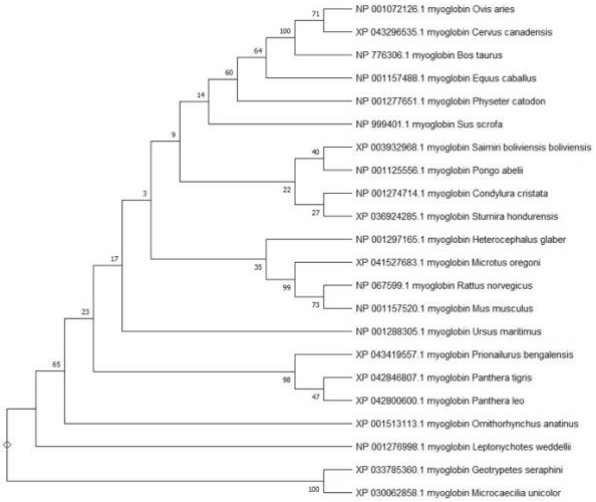
Muscle overcomes the greediness and inherent errors in progressive alignment by using an iterative approach. This works like progressive alignment (where two most closely related query sets align first and the next most closely related sequence is aligned to the previous alignment) except that the initial sequences are repeatedly realigned. Hence, MUSCLE being an iterative method improves upon progression methods with a more accurate distance measure to assess the relatedness between the two sequences. This distance measure is updated between the iterations. This enables muscle to be used even in case of distantly related sequences since the problem of gaps in consensus sequences and use of profile to compare sequences posed by progressive alignments is overcome.

5. Phylogeny (20 pts)

a) (5 pts) Construct a neighbor-joining tree of the 20 myoglobin protein sequences from problem #4, with 500 bootstrap replicates. Root your tree with the outgroup. Paste an image of the tree below with bootstrap values
The outgroups in this are Geotrypetes seraphini and Microcaecilia unicolor



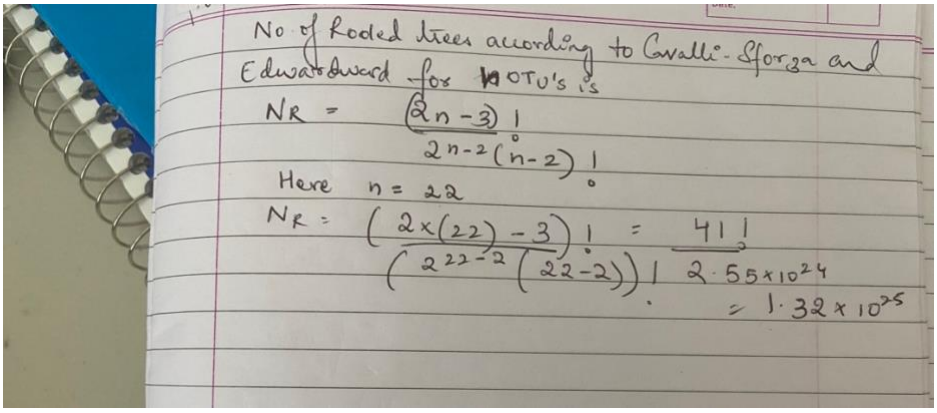
b) (5 pts) Construct a maximum likelihood tree of the same sequences and paste below. Does this tree have the same topology as the neighbor-joining tree? If not, what clades are different?



There are a few differences in the clades of the maximum likelihood and neighbor-joining trees. However, the overall topology represented by them remained similar. These were my observations in terms of differences in clade:

- **The node with bootstrap number as '14' illustrates inclusion of *Sus scrofa* with *bos taurus*, *equus caballus* and others. However, in the neighbor joining trees *Sus scrofa* is grouped with *Urus maritimus*. This was the only considerable change in the clade structure.**

c) (5 pts) What is the number of possible rooted trees for your sequences (21 or 22 including outgroup)?



d) (5 pts) What are key differences between neighbor-joining and maximum likelihood algorithms for phylogenetic reconstruction?

Neighbor-Joining	**Maximum Likelihood**
• This is a bottom-up clustering algorithm used for the, used for both nucleic acid and protein the algorithm requires knowledge of distance between each pair of taxa to form the tree (One quick tree)	• Infers phylogeny by evaluating a hypothesis in terms of the probability that a particular proposed model and the hypothesized history would give rise to the observed data set. It searches for tree with highest probability
• The algorithm is comparatively faster and not as CPU intensive • Result is not dependent on the model of evolution used	• CPU intensive and extremely slow • Result is dependent on the model of evolution used
• Good for analyzing large data sets and for bootstrapping	• Cannot be used for large scale data analysis or bootstrapping due to computational prohibition
• True tree with high probability can only be constructed when data of sufficient length is given	• Works accurately for sequences with missing data
• This is a distance-based method	• This is a character-based method