Project 3: Learning phylogenetic trees from multiple sequence alignment of the COVID data

Fereshteh Noroozi

Summer 2023

Data

I have downloaded the data for the Delta strain (number 5) from GISAID, covering the time from January 2021 to November 2022.

Data preparation

II. Align the collected dataset using the MAFFT tool and check the MSA for any standard format issues, such as duplicate taxon names, invalid characters in taxon names, or duplicate sequences.

Answer:

At first to align the collected dataset:

```
mafft fereshteh_gisaid_hcov-19_2023_06_27_10.fasta > fereshteh_gisaid_output.fasta
```

The output of the MAFFT software, which is commonly used for multiple sequence alignment, has the variables that are set to zero in the output:

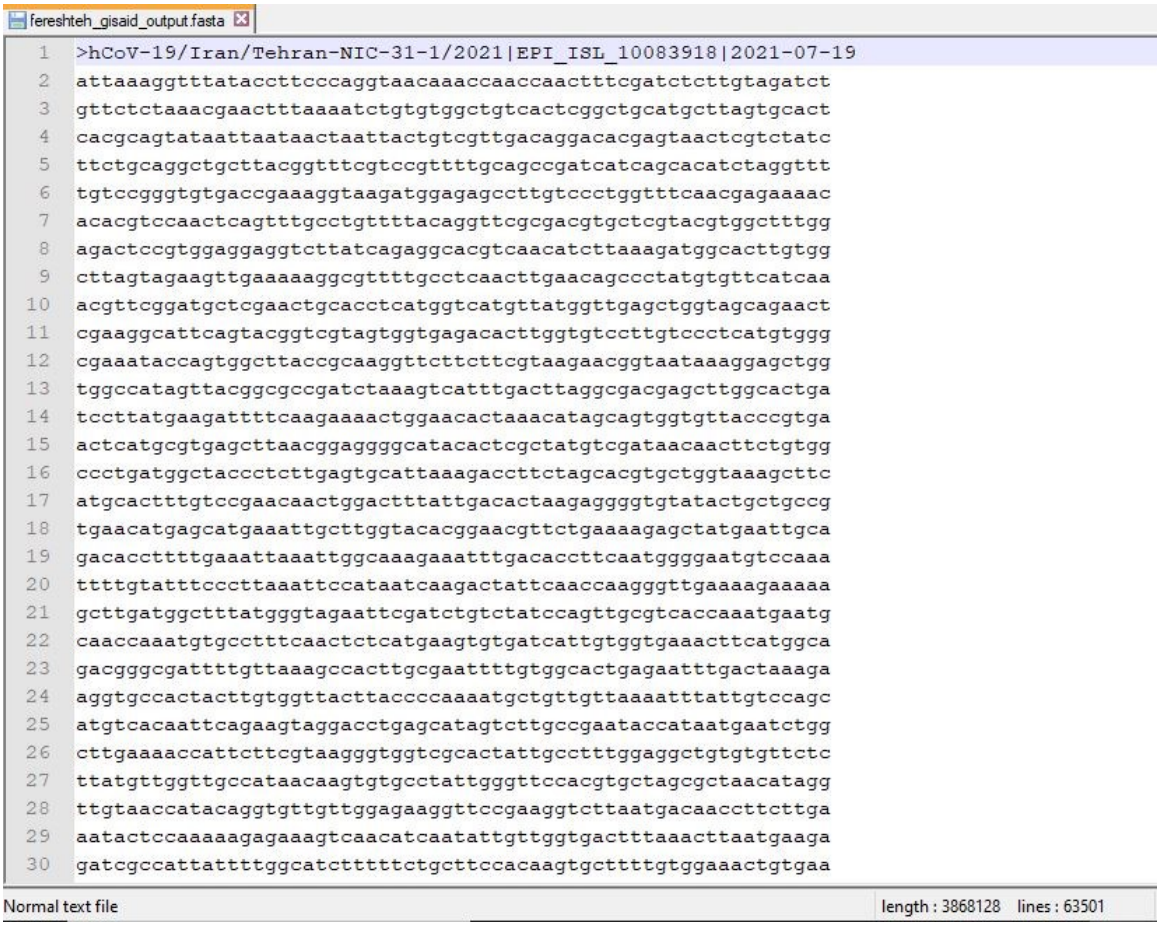
- nthread = 0

- nthreadpair = 0

- nthreadtb = 0

- ppenalty_ex = 0

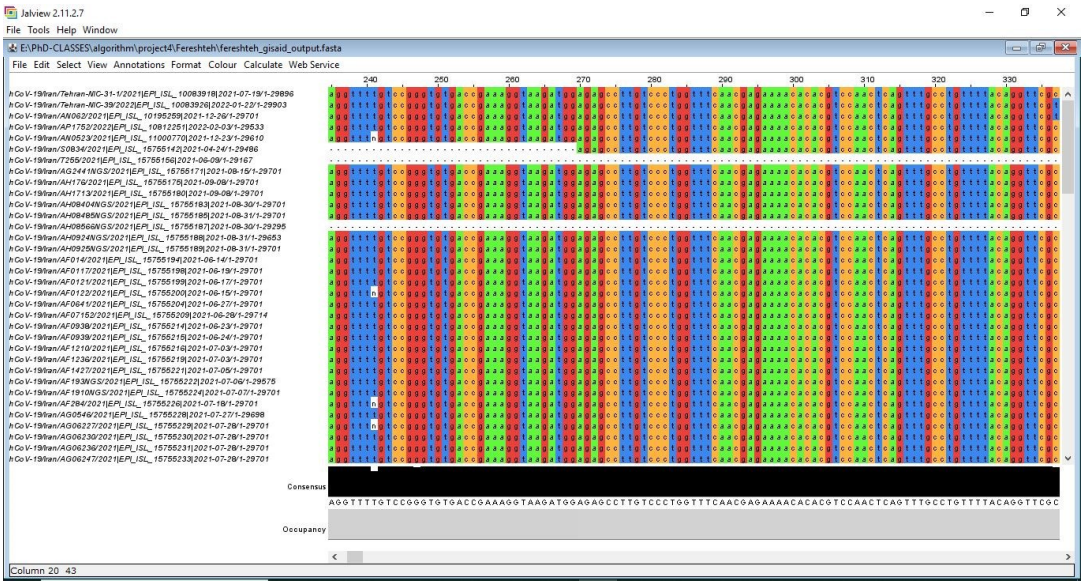
These variables might control the number of threads used for parallelization or some penalty values used in the alignment process. However, since they are set to zero, it seems that the default values are being used. The output also shows that the alignment is being performed using the FFT-NS-2 strategy with the DNA200 model and a gap penalty of -1.53 for opening a gap and 0 for extending a gap. The UPGMA algorithm is used for constructing guide trees, and the alignment is performed twice in a progressive manner.


fereshteh_gisaid_output.fasta

```
>hCoV-19/Iran/Tehran-NIC-31-1/2021|EPI_ISL_10083918|2021-07-19
attaaaggtttataccttcccaggtaacaaaccaaccaactttcgatctcttgtagatct
gttctctaaacgaactttaaaatctgtgtggctgtcactcggctgcatgcttagtgcact
cacgcagtataattaataactaattactgtcgttgacaggacacgagtaactcgtctatc
ttctgcaggctgcttacggtttcgtccgttttgcagccgatcatcagcacatctaggttt
tgtccgggtgtgaccgaaaggtaagatggagagccttgtccctggtttcaacgagaaaac
acacgtccaactcagtttgcctgttttacaggttcgcgacgtgctcgtacgtggctttgg
agactccgtggaggaggtcttatcagaggcacgtcaacatcttaaagatggcacttgtgg
cttagtagaagttgaaaaaggcgttttgcctcaacttgaacagccctatgtgttcatcaa
acgttcggatgctcgaactgcacctcatggtcatgttatggttgagctggtagcagaact
cgaaggcattcagtacggtcgtagtggtgagacacttggtgtccttgtccctcatgtggg
cgaaataccagtggcttaccgcaaggttcttcttcgtaagaacggtaataaaggagctgg
tggccatagttacggcgccgatctaaagtcatttgacttaggcgacgagcttggcactga
tccttatgaagattttcaagaaaactggaacactaaacatagcagtggtgttacccgtga
actcatgcgtgagcttaacggaggggcatacactcgctatgtcgataacaacttctgtgg
ccctgatggctaccctcttgagtgcattaaagaccttctagcacgtgctggtaaagcttc
atgcactttgtccgaacaactggactttattgacactaagaggggtgtatactgctgccg
tgaacatgagcatgaaattgcttggtacacggaacgttctgaaaagagctatgaattgca
gacacctttgaaattaaattggcaaagaaatttgacaccttcaatggggaatgtccaaa
ttttgtatttcccttaaattccataatcaagactattcaaccaagggttgaaaagaaaaa
gcttgatggctttatgggtagaattcgatctgtctatccagttgcgtcaccaaatgaatg
caaccaaatgtgcctttcaactctcatgaagtgtgatcattgtggtgaaacttcatggca
gacgggcgattttgttaaagccacttgcgaattttgtggcactgagaatttgactaaaga
aggtgccactacttgtggttacttaccccaaaatgctgttgttaaaatttattgtccagc
atgtcacaattcagaagtaggacctgagcatagtcttgccgaataccataatgaatctgg
cttgaaaaccattcttcgtaagggtggtcgcactattgcctttggaggctgtgtgttctc
ttatgttggttgccataacaagtgtgcctattgggttccacgtgctagcgctaacatagg
ttgtaaccatacaggtgttgttggagaaggttccgaaggtcttaatgacaaccttcttga
aatactccaaaaagagaaagtcaacatcaatattgttggtgactttaaacttaatgaaga
gatcgccattattttggcatctttttctgcttccacaagtgcttttgtggaaactgtgaa
```

Normal text file | length : 3868128 lines : 63501

Then I employed the Jalview software to visualize the alignment file.



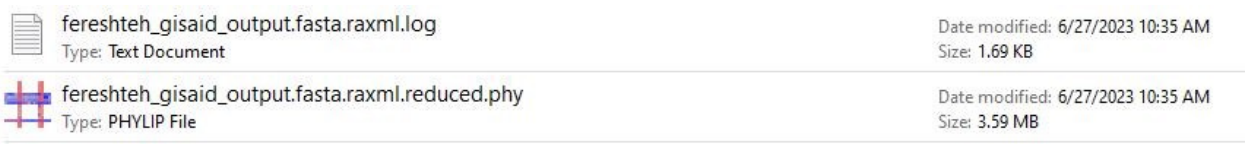
The colored regions represent the bases that have a match, while the dots indicate the positions where there is a mismatch

Subsequently, to ensure the MSA is free from standard format issues, such as duplicate taxon names, invalid characters in taxon names, or duplicate sequences, was checked.

```
raxml-ng --check --msa fereshteh_gisaid_output.fasta --model GTR+G
```

The output shows that the alignment file was successfully loaded with 127 taxa (sequences) and 29903 sites (positions) and that the alignment comprises one partition with a GTR+FO+G4m model. The percentage of gaps and invariant sites in the alignment is also shown (Alignment sites: 29903 ,Gaps: 2.04% , Invariant sites: 97.96%. This means that out of the 29903 alignment sites, 2.04% are gaps, and 97.96% are invariant (i.e., have the same nucleotide in all sequences).). The warning message "Duplicate sequences found: 1" indicates that there is one sequence that is identical to another sequence in the alignment.

This command produced two files:



1. fereshteh_gisaid_output.fasta.raxml.reduced.phy: This file is a reduced version of the input alignment file "fereshteh_gisaid_output.fasta", with duplicates and gap-only sites/taxa removed.
2. fereshteh_gisaid_output.fasta.raxml.log: This file contains the execution log of the RAxML-NG analysis. The log file may contain additional information about the analysis, such as the likelihood score, branch lengths, and tree topology.

Phylogenetic analysis

Estimate the optimal nucleotide substitution model using maximum likelihood or Bayesian methods by selecting the model with the lowest AIC or BIC score (you may use phangorn package in R).

Answer:

First, the code reads in the alignment data in PHYLIP format using the read.phyDat function in phangorn. Then, the modelTest function is used to estimate the optimal model using maximum likelihood and the AIC or BIC criterion. The which.min function is used to find the model with the lowest AIC and BIC scores, and the selected model name is stored in the best_model_AIC or best_model_BIC variable, respectively. Finally, the code prints out the name of the model with the lowest AIC and BIC score, allowing for the selection of the optimal model. Based on the output, the optimal nucleotide substitution model estimated using the modelTest function in phangorn and selected based on the lowest AIC and BIC scores is the General Time Reversible (GTR) model with gamma-distributed rate variation and a proportion of invariable sites (I), denoted as GTR+G(4)+I. The GTR+G(alpha)+I model is a popular choice for DNA sequence analyses, especially when the sequences under study have different base compositions or exhibit rate heterogeneity across sites. The model provides a good balance between model complexity and accuracy, and has been shown to perform well in a variety of phylogenetic contexts. The GTR model is often extended to account for rate heterogeneity across sites, which can be modeled by a gamma distribution with a shape parameter alpha and a rate heterogeneity parameter, denoted as GTR+G(alpha). Additionally, the model can account for the presence of invariable sites (i.e., sites that have no substitutions) by including a proportion of such sites, denoted as I, resulting in the GTR+G(alpha)+I model.

```r
# Load necessary packages
library(phangorn)
```

```
## Warning: package 'phangorn' was built under R version 4.2.3
```

```
## Loading required package: ape
```

```
## Warning: package 'ape' was built under R version 4.2.3
```

```r
# Read in the alignment data in PHYLIP format
alignment <- read.phyDat("fereshteh gisaid output.fasta.raxml.reduced.phy", format = "phylip")

# Estimate the optimal nucleotide substitution model using ML and AIC criterion
ml_model <- modelTest(alignment)
```

```
## Model         df  logLik    AIC      BIC
##            JC 249 -49653.41 99804.81 101872.9
##          JC+I 250 -49302.13 99104.26 101180.7
##       JC+G(4) 250 -49451.21 99402.41 101478.8
##     JC+G(4)+I 251 -49281.93 99065.86 101150.6
##           F81 252 -48679.13 97862.26 99955.3
##         F81+I 253 -48343.76 97193.53 99294.87
##      F81+G(4) 253 -48479.45 97464.9 99566.25
##    F81+G(4)+I 254 -48323.2 97154.4 99264.05
##           K80 250 -49436.51 99373.03 101449.5
##         K80+I 251 -49084.94 98671.88 100756.6
##      K80+G(4) 251 -49234.27 98970.54 101055.3
##    K80+G(4)+I 252 -49064.64 98633.29 100726.3
##           HKY 253 -48444.06 97394.11 99495.46
##         HKY+I 254 -48125.12 96758.24 98867.89
##      HKY+G(4) 254 -48247.15 97002.31 99111.96
##    HKY+G(4)+I 255 -48104.05 96718.09 98836.05
##           TrNe 251 -49349.06 99200.11 101284.8
##         TrNe+I 252 -49083 98510 100603
##      TrNe+G(4) 252 -49146.4 98796.8 100889.8
##    TrNe+G(4)+I 253 -48980.71 98467.42 100568.8
##           TrN 254 -48356.18 97220.36 99330.01
##         TrN+I 255 -48048.05 96606.11 98724.07
##      TrN+G(4) 255 -48160.05 96830.1 98948.05
##    TrN+G(4)+I 256 -48024.46 96560.92 98687.18
##          TPM1 251 -49375.83 99253.66 101338.4
##        TPM1+I 252 -49024.25 98552.5 100645.5
##     TPM1+G(4) 252 -49173.58 98851.16 100944.2
##   TPM1+G(4)+I 253 -49003.94 98513.88 100615.2
##          K81 251 -49375.83 99253.66 101338.4
##        K81+I 252 -49024.25 98552.5 100645.5
##     K81+G(4) 252 -49173.58 98851.16 100944.2
##   K81+G(4)+I 253 -49003.94 98513.88 100615.2
##         TPM1u 254 -48373.58 97255.15 99364.8
##       TPM1u+I 255 -48061.47 96632.95 98750.91
##    TPM1u+G(4) 255 -48177.81 96865.61 98983.57
## TPM1u+G(4)+I 256 -48040.2 96592.39 98718.66
##          TPM2 251 -49375.54 99253.08 101337.8
##        TPM2+I 252 -49037.27 98578.54 100671.6
##     TPM2+G(4) 252 -49175.31 98854.61 100947.7
##   TPM2+G(4)+I 253 -49016.88 98539.76 100641.1
##         TPM2u 254 -48363.72 97235.45 99345.1
##       TPM2u+I 255 -48053.46 96616.91 98734.87
##    TPM2u+G(4) 255 -48167.92 96845.84 98963.8
## TPM2u+G(4)+I 256 -48032.19 96576.38 98702.64
##          TPM3 251 -49423.18 99348.35 101433.1
##        TPM3+I 252 -49053.65 98611.3 100704.3
##     TPM3+G(4) 252 -49218.76 98941.52 101034.6
##   TPM3+G(4)+I 253 -49033.86 98573.72 100675.1
##         TPM3u 254 -48439.53 97387.06 99496.71
##       TPM3u+I 255 -48112.5 96734.99 98852.95
##    TPM3u+G(4) 255 -48241.59 96993.18 99111.14
## TPM3u+G(4)+I 256 -48091.81 96695.61 98821.88
##          TIM1e 252 -49288.37 99080.74 101173.8
##        TIM1e+I 253 -48942.31 98390.62 100492
##     TIM1e+G(4) 253 -49085.71 98677.43 100778.8
##   TIM1e+G(4)+I 254 -48920.02 98348.03 100457.7
##          TIM1 255 -48285.7 97081.4 99199.36
##        TIM1+I 256 -47983.76 96479.53 98605.79
##     TIM1+G(4) 256 -48090.7 96693.39 98819.66
##   TIM1+G(4)+I 257 -47960.13 96434.26 98568.82
##          TIM2e 252 -49288.08 99080.17 101173.2
##        TIM2e+I 253 -48952.12 98410.23 100511.6
##     TIM2e+G(4) 253 -49086.96 98679.92 100781.3
##   TIM2e+G(4)+I 254 -48930.39 98368.78 100478.4
##          TIM2 255 -48276.05 97062.09 99180.05
##        TIM2+I 256 -47976.64 96465.28 98591.54
##     TIM2+G(4) 256 -48081.51 96675.06 98801.32
##   TIM2+G(4)+I 257 -47953.78 96421.56 98556.13
##          TIM3e 252 -49335.72 99175.44 101268.5
##        TIM3e+I 253 -48971.49 98448.98 100550.3
##     TIM3e+G(4) 253 -49130.56 98767.13 100868.5
##   TIM3e+G(4)+I 254 -48958.12 98408.23 100517.9
##          TIM3 255 -48351.71 97213.42 99331.38
##        TIM3+I 256 -48037.05 96586.09 98712.36
##     TIM3+G(4) 256 -48154.72 96821.43 98947.69
##   TIM3+G(4)+I 257 -48014.08 96542.16 98676.73
##          TVMe 253 -49129.01 99164.01 101265.4
##        TVMe+I 254 -48971.72 98451.44 100561.1
##     TVMe+G(4) 254 -49126.02 98760.05 100869.7
##   TVMe+G(4)+I 255 -48951.65 98413.3 100531.3
##          TVM 256 -48323.86 97159.72 99285.99
##        TVM+I 257 -48006.87 96527.75 98662.41
##     TVM+G(4) 257 -48127.04 96768.08 98902.65
##   TVM+G(4)+I 258 -47985.76 96487.51 98630.38
##          SYM 254 -49241.56 98991.12 101100.8
##        SYM+I 255 -48885.74 98281.49 100399.4
##     SYM+G(4) 255 -49037.34 98584.68 100702.6
##   SYM+G(4)+I 256 -48864.91 98241.82 100368.1
##          GTR 257 -48236.24 96986.48 99121.05
##        GTR+I 258 -47931.34 96378.68 98521.55
##     GTR+G(4) 258 -48040.88 96597.75 98740.63
##   GTR+G(4)+I 259 -47908.97 96335.94 98487.12
```

```r
# Find the model with the lowest AIC score
best_model_AIC <- ml_model[which.min(ml_model$AIC), "Model"]

# Find the model with the lowest BIC score
best_model_BIC <- ml_model[which.min(ml_model$BIC), "Model"]


# Print the results
cat("lowest AIC score is", best_model_AIC, "\n")
```

```
## lowest AIC score is GTR+G(4)+I
```

```r
cat("lowest BIC score is", best_model_BIC, "\n")
```

```
## lowest BIC score is GTR+G(4)+I
```

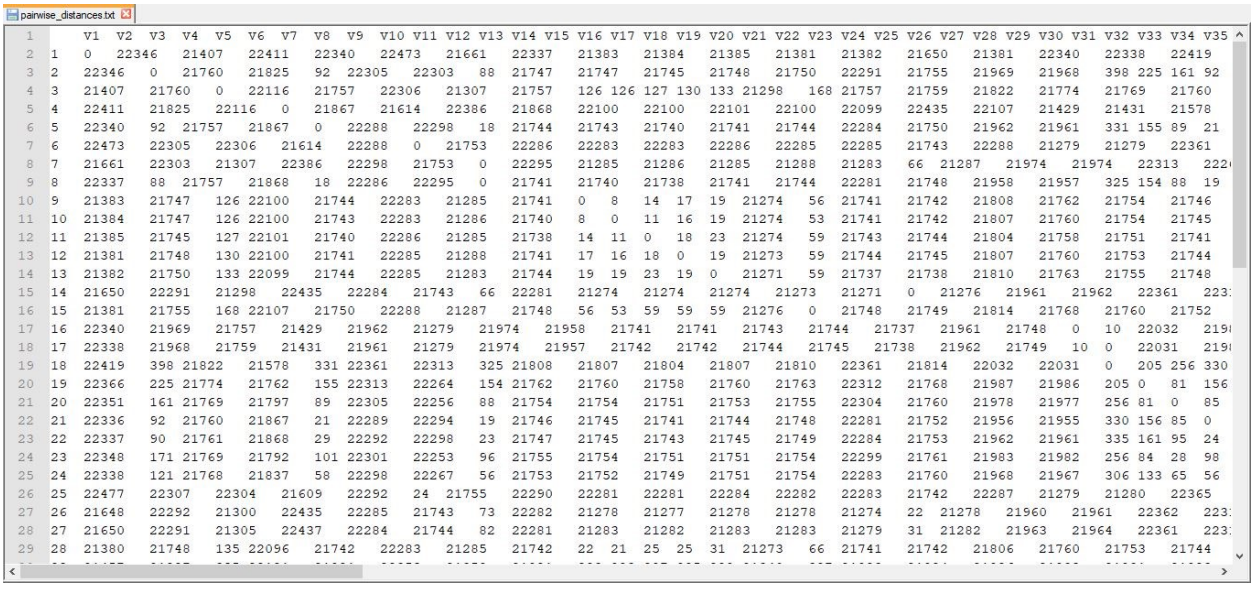
Calculate the pairwise distance between the sequences in the alignment.

Answer:

```
# Read in the alignment data in FASTA format
alignment <- readLines("fereshteh_gisaid_output.fasta.raxml.reduced.phy")

# Remove the header lines
alignment <- alignment[-seq(1, length(alignment), 2)]

# Create an empty distance matrix
num_seqs <- length(alignment)
dist_matrix <- matrix(0, nrow = num_seqs, ncol = num_seqs)

# Define a function to compute the distance between two sequences
pairwise_distance <- function(seq1, seq2) {
  num_diffs <- sum(charToRaw(seq1) != charToRaw(seq2))
  return(num_diffs)
}

# Calculate the pairwise distance matrix
for (i in 1:(num_seqs - 1)) {
  for (j in (i + 1):num_seqs) {
    dist_matrix[i, j] <- pairwise_distance(alignment[i], alignment[j])
    dist_matrix[j, i] <- dist_matrix[i, j]
  }
}
```

```
# Save the pairwise distance matrix to a text file
write.table(dist_matrix, "pairwise_distances.txt", sep = "\t", quote = FALSE, col.names = NA)
```

This R code calculates the pairwise distance between the sequences in the alignment. It does so by reading in the alignment data in FASTA format, removing the header lines, and then creating an empty distance matrix. It defines a function pairwise_distance to compute the distance between two sequences, and then calculates the pairwise distance matrix by looping through all pairs of sequences and applying the pairwise_distance function. Finally, it saves the resulting pairwise distance matrix to a text file.

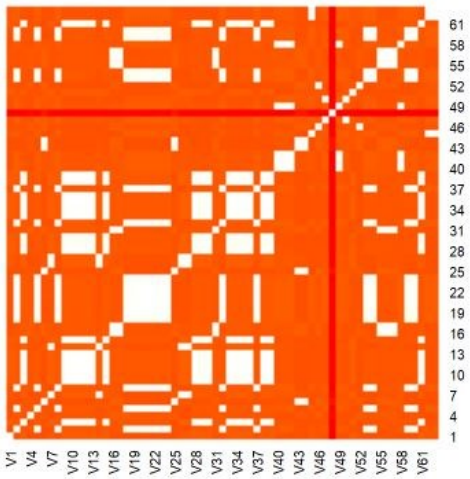


pairwise_distances.txt

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	V29	V30	V31	V32	V33	V34	V35
1	1	0	22346	21407	22411	22340	22473	21661	22337	21383	21384	21385	21381	21382	21650	21381	22340	22338	22419																
2	2	22346	0	21760	21825	92	22305	22303	88	21747	21747	21745	21748	21750	22291	21755	21969	21968	398	225	161	92													
3	3	21407	21760	0	22116	21757	22306	21307	21757	126	126	127	130	133	21298	168	21757	21759	21822	21774	21769	21760													
4	4	22411	21825	22116	0	21867	21614	22386	21868	22100	22100	22101	22100	22099	22435	22107	21429	21431	21578																
5	5	22340	92	21757	21867	0	22288	22298	18	21744	21743	21740	21741	21744	22284	21750	21962	21961	331	155	89	21													
6	6	22473	22305	22306	21614	22288	0	21753	22286	22283	22283	22286	22285	22285	21743	22288	21279	21279	22361																
7	7	21661	22303	21307	22386	22298	21753	0	22295	21285	21286	21285	21288	21283	66	21287	21974	21974	22313	222															
8	8	22337	88	21757	21868	18	22286	22295	0	21741	21740	21738	21741	21744	22281	21748	21958	21957	325	154	88	19													
9	9	21383	21747	126	22100	21744	22283	21285	21741	0	8	14	17	19	21274	56	21741	21742	21808	21762	21754	21746													
10	10	21384	21747	126	22100	21743	22283	21286	21740	8	0	11	16	19	21274	53	21741	21742	21807	21760	21754	21745													
11	11	21385	21745	127	22101	21740	22286	21285	21738	14	11	0	18	23	21274	59	21743	21744	21804	21758	21751	21741													
12	12	21381	21748	130	22100	21741	22285	21288	21741	17	16	18	0	19	21273	59	21744	21745	21807	21760	21753	21744													
13	13	21382	21750	133	22099	21744	22285	21283	21744	19	19	23	19	0	21271	59	21737	21738	21810	21763	21755	21748													
14	14	21650	22291	21298	22435	22284	21743	66	22281	21274	21274	21274	21273	21271	0	21276	21961	21962	22361	223															
15	15	21381	21755	168	22107	21750	22288	21287	21748	56	53	59	59	59	21276	0	21748	21749	21814	21768	21760	21752													
16	16	22340	21969	21757	21429	21962	21279	21974	21958	21741	21741	21743	21744	21737	21961	21748	0	10	22032	219															
17	17	22338	21968	21759	21431	21961	21279	21974	21957	21742	21742	21744	21745	21738	21962	21749	10	0	22031	219															
18	18	22419	398	21822	21578	331	22361	22313	325	21808	21807	21804	21807	21810	22361	21814	22032	22031	0	205	256	330													
19	19	22366	225	21774	21762	155	22313	22264	154	21762	21760	21758	21760	21763	22312	21768	21987	21986	205	0	81	156													
20	20	22351	161	21769	21797	89	22305	22256	88	21754	21754	21751	21753	21755	22304	21760	21978	21977	256	81	0	85													
21	21	22336	92	21760	21867	21	22289	22294	19	21746	21745	21741	21744	21748	22281	21752	21956	21955	330	156	85	0													
22	22	22337	90	21761	21868	29	22292	22298	23	21747	21745	21743	21745	21749	22284	21753	21962	21961	335	161	95	24													
23	23	22348	171	21769	21792	101	22301	22253	96	21755	21754	21751	21751	21754	22299	21761	21983	21982	256	84	28	98													
24	24	22338	121	21768	21837	58	22298	22267	56	21753	21752	21749	21751	21754	22283	21760	21968	21967	306	133	65	56													
25	25	22477	22307	22304	21609	22292	24	21755	22290	22281	22281	22284	22282	22283	21742	22287	21279	21280	22365																
26	26	21648	22292	21300	22435	22285	21743	73	22282	21278	21277	21278	21278	21274	22	21278	21960	21961	22362	223															
27	27	21650	22291	21305	22437	22284	21744	82	22281	21283	21282	21283	21283	21279	31	21282	21963	21964	22361	223															
28	28	21380	21748	135	22096	21742	22283	21285	21742	22	21	25	25	31	21273	66	21741	21742	21806	21760	21753	21744													

A heatmap is a good choice if we want to visualize the overall structure of the distance matrix and identify patterns or clusters. In a heatmap, the matrix is represented as a grid of colored squares, with darker colors indicating higher distances. We can also add clustering to the rows and/or columns of the heatmap to group similar points together.

```
# Read in the pairwise distance matrix
dist_matrix <- as.matrix(read.table("pairwise_distances.txt"))

# Create a heatmap of the distance matrix
heatmap(dist_matrix, Rowv = NA, Colv = NA, col = rev(heat.colors(256)),
        scale = "none", margins = c(10, 10))
```



Generate a first tree topology based on pairwise distances.

Answer:

```
# Load necessary packages
library(ape)
```

```
## Warning: package 'ape' was built under R version 4.2.3
```

```
# Read in the pairwise distance matrix
dist_matrix <- as.matrix(read.table("pairwise_distances.txt", header = FALSE, fill = TRUE))

# Perform the neighbor-joining algorithm
tree <- nj(dist_matrix)
```

```
## Warning in storage.mode(m) <- "numeric": NAs introduced by coercion
```

```
# Save the tree topology to a file
write.tree(tree, "pairwise_distances.nwk")
```

This code performs a neighbor-joining algorithm on the matrix using the nj() function from the ape package, and saves the resulting tree topology to a Newick file "pairwise_distances.nwk" using the write.tree() function.

Neighbour-joining (NJ) is an algorithm for building phylogenetic trees based on pairwise distances between taxa. The basic idea behind the NJ algorithm is to iteratively join pairs of taxa that are the "closest" in terms of their pairwise distances, until a complete tree is constructed.

pairwise_distances.nwk

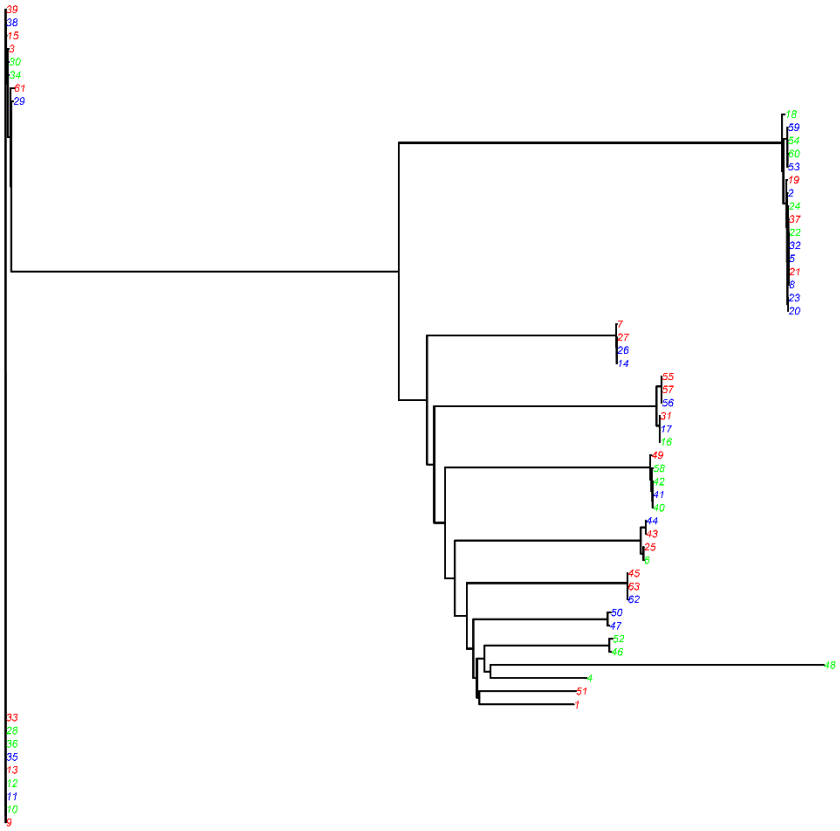
```
((V36:10.13043229,V37:10.86956771):1.585071087,(V29:13.19999371,V34:12.80000629):1.414928913,(((((V11:3.604402184,V12:7.395597816):0.8161770276,V
```

Then to plot my tree:

```
# Load the ape package
library(ape)
```

```
## Warning: package 'ape' was built under R version 4.2.3
```

```
# Read in the pairwise distance matrix
dist_matrix <- as.matrix(read.table("pairwise_distances.txt"))

# Convert the distance matrix to a distance object
dist_obj <- dist(dist_matrix)

# Generate a tree topology using the neighbor-joining algorithm
tree <- nj(dist_obj)

# Define a vector of colors for each group
group_colors <- c("red", "blue", "green")

# Assign a color to each tip label based on the group variable
group <- c("Group1", "Group2", "Group1", "Group3", "Group2", "Group3")
tip_colors <- group_colors[as.factor(group)]

# Plot the tree with larger font size, longer branches, and colored tip labels
plot(tree, cex = 0.3, edge.length = 5, tip.color = tip_colors)
```



Optimize the phylogenetic tree topology.

Answer:

The bellow code optimizes the tree topology using maximum likelihood estimation (pml()) from the phangorn package. The model used is General Time Reversible (GTR) with gamma-distributed rate heterogeneity and an estimated proportion of invariable sites. The optInv, optGamma, and optParams options are set to TRUE to optimize the proportion of invariable sites, the shape parameter of the gamma distribution, and the base frequencies, respectively. The trace option is set to 0 to suppress the output of optimization details.

```r
# Load necessary packages
library(ape)
```

```
## Warning: package 'ape' was built under R version 4.2.3
```

```r
library(phangorn)
```

```
## Warning: package 'phangorn' was built under R version 4.2.3
```

```r
# Read in the nucleotide alignment
alignment <- read.phyDat("fereshteh_gisaid_output.fasta.raxml.reduced.phy", format = "phylip")

# Convert the alignment to a DNAbin object
alignment_bin <- as.DNAbin(alignment)

# Calculate pairwise distances between sequences
distances <- dist.dna(alignment_bin, model = "raw")

# Generate a tree topology based on pairwise distances
tree <- nj(distances)

# Optimize the phylogenetic tree topology using ML estimation
fit <- pml(tree, data = alignment, model = "GTR", optInv = TRUE, optGamma = TRUE,
           optParams = TRUE, trace = 0)

# Save the optimized tree topology to a file
write.tree(fit$tree, "optimized_tree.nwk")
```

Then to plot my tree:

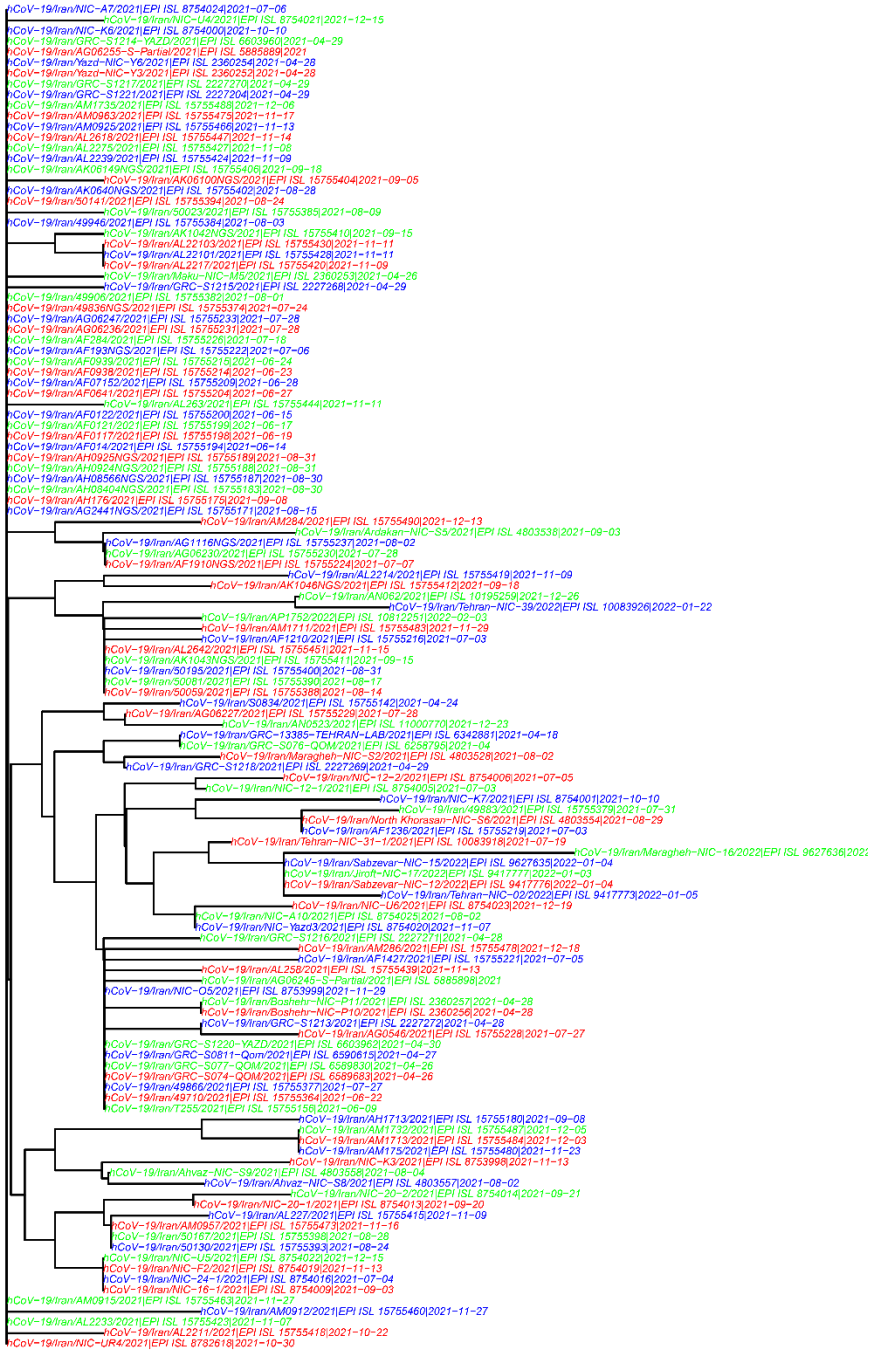
```r
# Load necessary packages
library(ape)
```

```
## Warning: package 'ape' was built under R version 4.2.3
```

```r
library(phangorn)
```

```
## Warning: package 'phangorn' was built under R version 4.2.3
```

```r
# Read in the optimized tree topology
tree_optimized <- read.tree("optimized_tree.nwk")

# Prune the tree to remove some nodes or branches
tree_pruned <- drop.tip(tree_optimized, c("hCoV-19/Iran/NIC-A7/2021|EPI_ISL_8754024|2021-07-06:1e-08"))

# Collapse multifurcating nodes to simplify the tree
tree_simple <- multi2di(tree_pruned)

# Define a vector of colors for each group
group_colors <- c("red", "blue", "green")

# Assign a color to each tip label based on the group variable
group <- c("Group1", "Group2", "Group1", "Group3", "Group2", "Group3")
tip_colors <- group_colors[as.factor(group)]

# Plot the simplified tree with larger font size, longer branches, and colored tip labels
plot(tree_simple, cex = 0.3, edge.length = 10, tip.color = tip_colors)
```



Optimize the branch lengths.

Answer:

The bellow code optimizes the branch lengths of the tree using maximum likelihood estimation (optim.pml()) from the phangorn package. This step refines the branch lengths of the tree based on the optimized topology. The optNni, optBf, and nniBf options are set to TRUE to optimize the

branch lengths using the NNI (nearest-neighbor interchange) and BF (branch swapping) algorithms. The trace option is set to 0 to suppress the output of optimization details.

```
# Load necessary packages
library(ape)
```

```
## Warning: package 'ape' was built under R version 4.2.3
```

```
library(phangorn)
```

```
## Warning: package 'phangorn' was built under R version 4.2.3
```

```
# Read in the nucleotide alignment
alignment <- read.phyDat("fereshteh_gisaid_output.fasta.raxml.reduced.phy", format = "phylip")

# Convert the alignment to a DNAbin object
alignment_bin <- as.DNAbin(alignment)

# Calculate pairwise distances between sequences
distances <- dist.dna(alignment_bin, model = "raw")

# Generate a tree topology based on pairwise distances
tree <- nj(distances)

# Optimize the phylogenetic tree topology using ML estimation
fit <- pml(tree, data = alignment, model = "GTR", optInv = TRUE, optGamma = TRUE,
           optParams = TRUE, trace = 0)

# Optimize the branch lengths using ML estimation
fit <- optim.pml(fit, optNni = TRUE, optBf = TRUE, nniBf = TRUE, trace = 0)
```

```
## optimize edge weights:  -56028.29 --> -51747.41
## optimize base frequencies:  -51747.41 --> -50796.31
## optimize edge weights:  -50796.31 --> -50796.11
## optimize topology:  -50796.11 --> -50173.84  NNI moves:  65
## optimize base frequencies:  -50173.84 --> -50173.8
## optimize edge weights:  -50173.8 --> -50173.8
## optimize topology:  -50173.8 --> -50085.55  NNI moves:  23
## optimize base frequencies:  -50085.55 --> -50085.55
## optimize edge weights:  -50085.55 --> -50085.55
## optimize topology:  -50085.55 --> -50043.02  NNI moves:  9
## optimize base frequencies:  -50043.02 --> -50043.02
## optimize edge weights:  -50043.02 --> -50043.02
## optimize topology:  -50043.02 --> -50033.04  NNI moves:  4
## optimize base frequencies:  -50033.04 --> -50033.04
## optimize edge weights:  -50033.04 --> -50033.04
## optimize topology:  -50033.04 --> -50033.04  NNI moves:  0
```

```
# Save the optimized tree topology and branch lengths to a file
write.tree(fit$tree, "optimized_tree_branch.nwk")
```

Then to plot my tree:

```r
# Load necessary packages
library(ape)
```

```
## Warning: package 'ape' was built under R version 4.2.3
```

```r
library(phangorn)
```

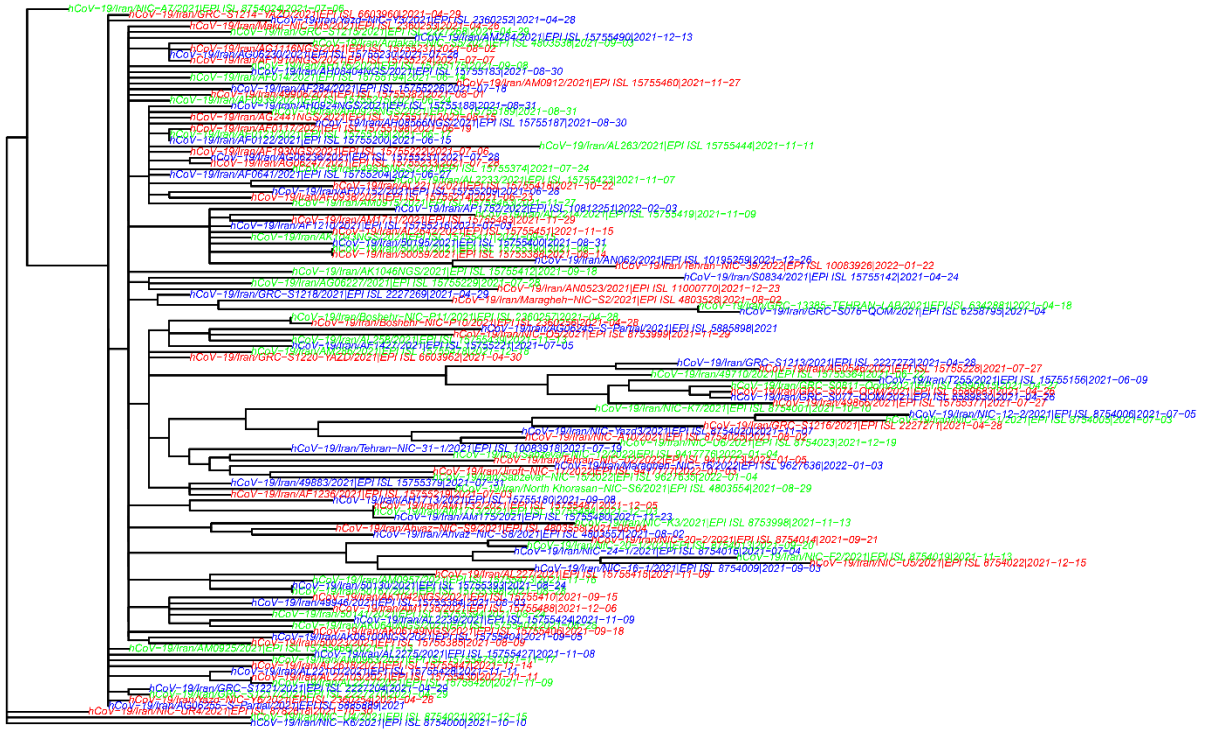
```
## Warning: package 'phangorn' was built under R version 4.2.3
```

```r
# Read in the optimized tree topology
tree_optimized <- read.tree("optimized_tree_branch.nwk")

# Prune the tree to remove some nodes or branches
tree_pruned <- drop.tip(tree_optimized, c("hCoV-19/Iran/NIC-A7/2021|EPI_ISL_8754024|2021-07-06:1e-08"))

# Collapse multifurcating nodes to simplify the tree
tree_simple <- multi2di(tree_pruned)

# Define a vector of colors for each group
group_colors <- c("red", "blue", "green")

# Assign a color to each tip label based on the group variable
group <- c("Group1", "Group2", "Group1", "Group3", "Group2", "Group3")
tip_colors <- group_colors[as.factor(group)]

# Plot the simplified tree with larger font size, longer branches, and colored tip labels
plot(tree_simple, cex = 0.3, edge.length = 10, tip.color = tip_colors)
```

```
## Warning in plot.window(...): "edge.length" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "edge.length" is not a graphical parameter
```

```
## Warning in title(...): "edge.length" is not a graphical parameter
```



Optimize the nucleotide substitution model.

Answer:

The bellow code, optimizes the nucleotide substitution model parameters for the tree using maximum likelihood estimation (pml()) from the phangorn package. The model used is the best-fitting model selected in step 3. The optInv, optGamma, and optParams options are set to TRUE to optimize the proportion of invariable sites, the shape parameter of the gamma distribution, and the base frequencies, respectively. The trace option is set to 0 to suppress the output of optimization details.

```r
# Load necessary packages
library(ape)
```

```
## Warning: package 'ape' was built under R version 4.2.3
```

```r
library(phangorn)
```

```
## Warning: package 'phangorn' was built under R version 4.2.3
```

```r
# Read in the nucleotide alignment
alignment <- read.phyDat("fereshteh gisaid output.fasta.raxml.reduced.phy", format = "phylip")

# Estimate the best-fitting nucleotide substitution model
model <- modelTest(alignment)
```

```
## Model        df  LogLik   AIC      BIC
##           JC 249 -49653.41 99804.81 101872.9
##         JC+I 250 -49302.13 99104.26 101180.7
##       JC+G(4) 250 -49451.21 99402.41 101478.8
##     JC+G(4)+I 251 -49281.93 99065.86 101150.6
##           F81 252 -48679.13 97862.26 99955.3
##         F81+I 253 -48343.76 97193.53 99294.87
##       F81+G(4) 253 -48479.45 97464.9 99568.25
##     F81+G(4)+I 254 -48323.2 97154.4 99264.05
##           K80 250 -49436.51 99373.03 101449.5
##         K80+I 251 -49084.94 98671.88 100756.6
##       K80+G(4) 251 -49234.27 98970.54 101055.3
##     K80+G(4)+I 252 -49054.54 98633.29 100776.3
##           HKY 253 -48444.06 97394.11 99495.46
##         HKY+I 254 -48125.12 96758.24 98867.89
##       HKY+G(4) 254 -48247.15 97002.31 99111.96
##     HKY+G(4)+I 255 -48104.05 96718.09 98836.05
##         TrNe 251 -49349.06 99200.11 101284.8
##       TrNe+I 252 -49003 98510 100503
##     TrNe+G(4) 252 -49146.4 98796.8 100889.8
##   TrNe+G(4)+I 253 -48980.71 98467.42 100568.8
##           TrN 254 -48356.18 97220.36 99330.01
##         TrN+I 255 -48048.05 96606.11 98724.07
##       TrN+G(4) 255 -48160.05 96830.1 98948.05
##     TrN+G(4)+I 256 -48024.46 96560.92 98687.18
##         TPM1 251 -49375.83 99253.66 101338.4
##       TPM1+I 252 -49024.25 98552.5 100645.5
##     TPM1+G(4) 252 -49173.58 98851.16 100944.2
##   TPM1+G(4)+I 253 -49003.94 98513.88 100615.2
##           K81 251 -49375.83 99253.66 101338.4
##         K81+I 252 -49024.25 98552.5 100645.5
##       K81+G(4) 252 -49173.58 98851.16 100944.2
##     K81+G(4)+I 253 -49003.94 98513.88 100615.2
##         TPM1u 254 -48373.58 97255.15 99364.8
##       TPM1u+I 255 -48061.47 96632.95 98750.91
##     TPM1u+G(4) 255 -48177.81 96865.61 98983.57
##   TPM1u+G(4)+I 256 -48040.2 96592.39 98718.66
##         TPM2 251 -49375.54 99253.08 101337.8
##       TPM2+I 252 -49037.27 98578.54 100671.6
##     TPM2+G(4) 252 -49175.31 98854.63 100947.7
##   TPM2+G(4)+I 253 -49016.88 98539.76 100641.1
##         TPM2u 254 -48363.72 97235.45 99345.1
##       TPM2u+I 255 -48053.46 96616.91 98734.87
##     TPM2u+G(4) 255 -48167.92 96845.84 98963.8
##   TPM2u+G(4)+I 256 -48032.19 96576.38 98702.64
##         TPM3 251 -49423.18 99348.35 101433.1
##       TPM3+I 252 -49053.65 98611.3 100704.3
##     TPM3+G(4) 252 -49218.76 98941.52 101034.6
##   TPM3+G(4)+I 253 -49033.86 98573.72 100675.1
##         TPM3u 254 -48439.53 97387.06 99496.71
##       TPM3u+I 255 -48112.5 96734.99 98852.95
##     TPM3u+G(4) 255 -48241.59 96993.18 99111.14
##   TPM3u+G(4)+I 256 -48091.81 96695.61 98821.88
##         TIM1e 252 -49288.37 99080.74 101173.8
##       TIM1e+I 253 -48942.31 98390.62 100492
##     TIM1e+G(4) 253 -49085.71 98677.43 100778.8
##   TIM1e+G(4)+I 254 -48920.02 98348.03 100457.7
##         TIM1 255 -48285.7 97081.4 99199.36
##       TIM1+I 256 -47983.76 96479.53 98605.79
##     TIM1+G(4) 256 -48090.7 96693.39 98819.66
##   TIM1+G(4)+I 257 -47950.13 96434.26 98568.82
##         TIM2e 252 -49288.08 99080.17 101173.2
##       TIM2e+I 253 -48952.12 98410.23 100511.6
##     TIM2e+G(4) 253 -49086.96 98679.92 100781.3
##   TIM2e+G(4)+I 254 -48930.39 98368.78 100478.4
##         TIM2 255 -48276.05 97062.09 99180.05
##       TIM2+I 256 -47976.64 96465.28 98591.54
##     TIM2+G(4) 256 -48081.53 96675.06 98801.32
##   TIM2+G(4)+I 257 -47953.78 96471.56 98556.13
##         TIM3e 252 -49335.72 99175.44 101268.5
##       TIM3e+I 253 -48971.49 98448.98 100550.3
##     TIM3e+G(4) 253 -49130.56 98767.11 100868.5
##   TIM3e+G(4)+I 254 -48950.12 98408.23 100517.9
##         TIM3 255 -48351.71 97213.42 99331.38
##       TIM3+I 256 -48037.05 96586.09 98712.36
##     TIM3+G(4) 256 -48154.72 96821.43 98947.69
##   TIM3+G(4)+I 257 -48014.08 96542.16 98676.73
##         TVMe 253 -49329.01 99164.03 101265.4
##       TVMe+I 254 -48971.72 98451.44 100561.1
##     TVMe+G(4) 254 -49126.02 98760.05 100869.7
##   TVMe+G(4)+I 255 -48951.65 98413.3 100531.3
##         TVM 256 -48323.86 97159.72 99285.99
##       TVM+I 257 -48006.87 96527.75 98662.31
##     TVM+G(4) 257 -48127.04 96768.08 98902.65
##   TVM+G(4)+I 258 -47985.76 96487.51 98630.38
##         SYM 254 -49241.56 98991.12 101100.8
##       SYM+I 255 -48885.74 98281.49 100399.4
##     SYM+G(4) 255 -49037.34 98584.68 100702.6
##   SYM+G(4)+I 256 -48864.91 98241.82 100368.1
##         GTR 257 -48236.24 96986.48 99121.05
##       GTR+I 258 -47931.34 96378.68 98521.55
##     GTR+G(4) 258 -48040.88 96597.75 98740.83
##   GTR+G(4)+I 259 -47908.97 96335.94 98487.12
```

```r
# Convert the alignment to a DNAbin object
alignment_bin <- as.DNAbin(alignment)

# Generate a tree object based on the optimized tree topology and branch lengths
tree <- read.tree("optimized_tree_branch.nwk")

# Optimize the nucleotide substitution model parameters using ML estimation
fit <- pml(tree, data = alignment, model = model, optInv = TRUE, optGamma = TRUE,
           optParams = TRUE, trace = 0)

# Save the optimized tree topology, branch lengths, and nucleotide substitution model parameters to a file
write.tree(fit$tree, "optimized_tree_model.nwk")
```

Then to plot my tree:

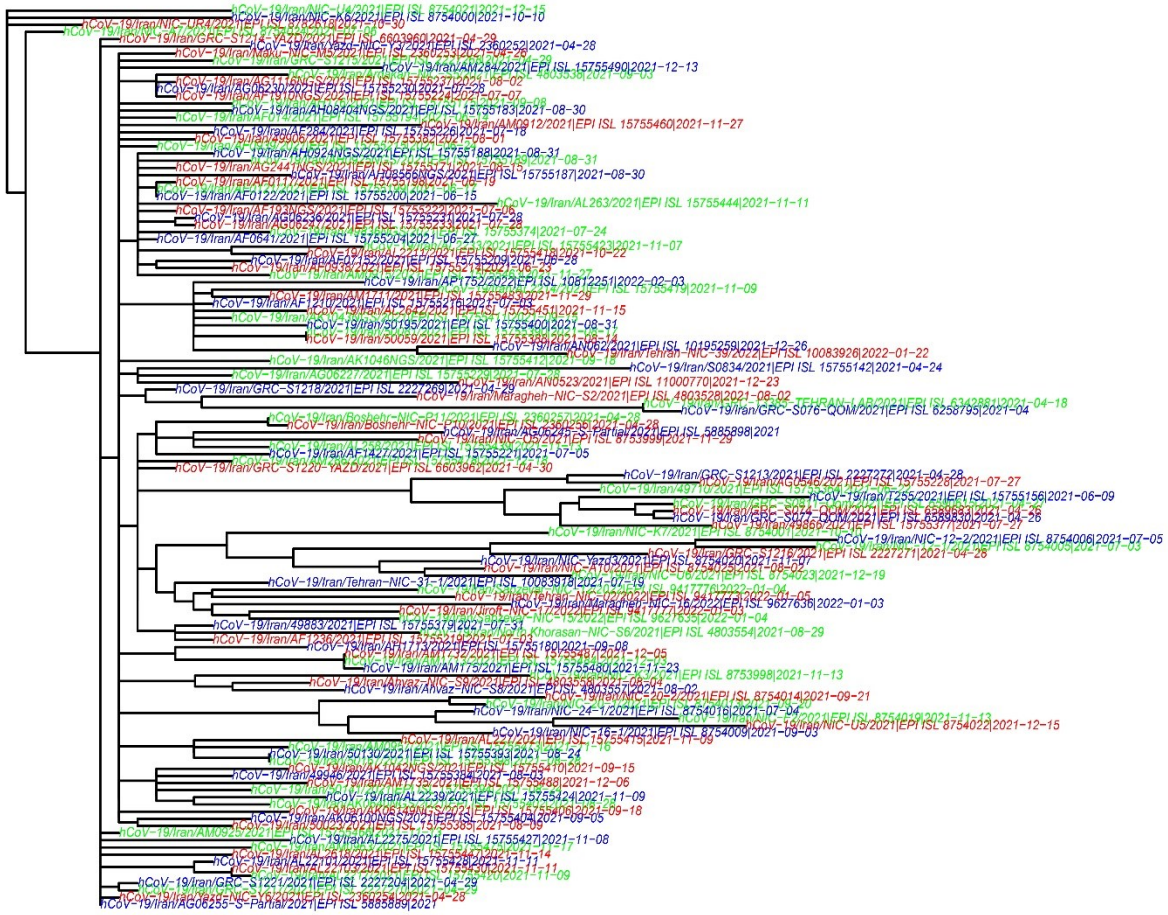
```
# Load necessary packages
library(ape)
```

```
## Warning: package 'ape' was built under R version 4.2.3
```

```
library(phangorn)
```

```
## Warning: package 'phangorn' was built under R version 4.2.3
```

```
# Read in the optimized tree topology
tree_optimized <- read.tree("optimized_tree_model.nwk")

# Prune the tree to remove some nodes or branches
tree_pruned <- drop.tip(tree_optimized, c("hCoV-19/Iran/NIC-A7/2021|EPI_ISL_8754024|2021-07-06:1e-08"))

# Collapse multifurcating nodes to simplify the tree
tree_simple <- multi2di(tree_pruned)

# Define a vector of colors for each group
group_colors <- c("red", "blue", "green")

# Assign a color to each tip label based on the group variable
group <- c("Group1", "Group2", "Group1", "Group3", "Group2", "Group3")
tip_colors <- group_colors[as.factor(group)]

# Plot the simplified tree with larger font size, longer branches, and colored tip labels
plot(tree_simple, cex = 0.3, edge.length = 10, tip.color = tip_colors)
```


hCoV-19/Iran/NIC-U4/2021|EPI ISL 8754021|2021-12-15
hCoV-19/Iran/NIC-K6/2021|EPI ISL 8754000|2021-10-10
hCoV-19/Iran/NIC-UR4/2021|EPI ISL 8754014|2021-10-30
hCoV-19/Iran/GRC-S1214-YAZD/2021|EPI ISL 6603960|2021-04-29
hCoV-19/Iran/Yazd-NIC-13/2021|EPI ISL 2360252|2021-04-28
hCoV-19/Iran/Kashan-NIC-MS2021|EPI ISL 2360253|2021-04-28
hCoV-19/Iran/NIC-S1214/2021|EPI ISL 8754009|2021-09-03
hCoV-19/Iran/AM284/2021|EPI ISL 15755490|2021-12-13
hCoV-19/Iran/AG0522/2021|EPI ISL 15755210|2021-06-02
hCoV-19/Iran/AG0522b/2021|EPI ISL 15755201|2021-06-28
hCoV-19/Iran/AF1910NGS/2021|EPI ISL 15755224|2021-07-07
hCoV-19/Iran/AH08404NGS/2021|EPI ISL 15755183|2021-08-30
hCoV-19/Iran/AM312/2021|EPI ISL 15755460|2021-11-27
hCoV-19/Iran/AF284/2021|EPI ISL 15755526|2021-11-18
hCoV-19/Iran/49906/2021|EPI ISL 15755382|2021-08-01
hCoV-19/Iran/AH0924NGS/2021|EPI ISL 15755188|2021-08-31
hCoV-19/Iran/AG244/2021|EPI ISL 15755191|2021-08-31
hCoV-19/Iran/AH0856NGS/2021|EPI ISL 15755187|2021-08-30
hCoV-19/Iran/AF0117/2021|EPI ISL 15755198|2021-06-19
hCoV-19/Iran/AF0122/2021|EPI ISL 15755200|2021-06-15
hCoV-19/Iran/AL263/2021|EPI ISL 15755444|2021-11-11
hCoV-19/Iran/AF1910NGS/2021|EPI ISL 15755225|2021-07-06
hCoV-19/Iran/AG0236/2021|EPI ISL 15755231|2021-07-28
hCoV-19/Iran/AG0624/2021|EPI ISL 15755234|2021-07-28
hCoV-19/Iran/AM374/2021|EPI ISL 15755374|2021-07-24
hCoV-19/Iran/AF0641/2021|EPI ISL 15755204|2021-06-27
hCoV-19/Iran/AL423/2021|EPI ISL 15755423|2021-11-07
hCoV-19/Iran/AF0132/2021|EPI ISL 15755209|2021-06-28
hCoV-19/Iran/AF0038/2021|EPI ISL 15755213|2021-06-23
hCoV-19/Iran/AP1752/2022|EPI ISL 10812251|2022-02-03
hCoV-19/Iran/AM1702/2021|EPI ISL 15755483|2021-11-29
hCoV-19/Iran/AF1210/2021|EPI ISL 15755219|2021-07-03
hCoV-19/Iran/AF0642/2021|EPI ISL 15755451|2021-11-15
hCoV-19/Iran/50195/2021|EPI ISL 15755400|2021-08-31
hCoV-19/Iran/50059/2021|EPI ISL 15755388|2021-08-24
hCoV-19/Iran/AN062/2021|EPI ISL 10195259|2021-12-26
hCoV-19/Iran/AK1046NGS/2021|EPI ISL 15755412|2021-09-18
hCoV-19/Iran/AG0522/2021|EPI ISL 15755209|2021-06-28
hCoV-19/Iran/GRC-S1278/2021|EPI ISL 2227269|2021-04-29
hCoV-19/Iran/AN0523/2021|EPI ISL 11000770|2021-12-23
hCoV-19/Iran/Maragheh-NIC-S2/2021|EPI ISL 4803528|2021-08-02
hCoV-19/Iran/Boshehr-NIC-P11/2021|EPI ISL 2360270|2021-04-28
hCoV-19/Iran/Boshehr-NIC-P10/2021|EPI ISL 2360256|2021-04-28
hCoV-19/Iran/NIC-OS/2021|EPI ISL 5885898|2021
hCoV-19/Iran/NIC-OS/2021|EPI ISL 8753998|2021-11-23
hCoV-19/Iran/AF1427/2021|EPI ISL 15755221|2021-07-05
hCoV-19/Iran/GRC-S1220-YAZD/2021|EPI ISL 6603962|2021-04-30
hCoV-19/Iran/GRC-S1213/2021|EPI ISL 2227272|2021-04-28
hCoV-19/Iran/AG0548/2021|EPI ISL 15755228|2021-07-27
hCoV-19/Iran/4970/2021|EPI ISL 15755156|2021-06-09
hCoV-19/Iran/T255/2021|EPI ISL 15755156|2021-06-09
hCoV-19/Iran/GRC-S074-QOM/2021|EPI ISL 6589830|2021-04-26
hCoV-19/Iran/NIC-K7/2021|EPI ISL 8754001|2021-10-10
hCoV-19/Iran/NIC-12-2/2021|EPI ISL 8754006|2021-07-05
hCoV-19/Iran/GRC-S1216/2021|EPI ISL 2227271|2021-04-28
hCoV-19/Iran/NIC-Yazd3/2021|EPI ISL 8754020|2021-11-07
hCoV-19/Iran/NIC-S10/2021|EPI ISL 8754023|2021-12-19
hCoV-19/Iran/Tehran-NIC-31-1/2021|EPI ISL 10083978|2021-07-19
hCoV-19/Iran/Tehran-NIC-02/2021|EPI ISL 9417223|2022-01-05
hCoV-19/Iran/Maragheh-NIC-16/2022|EPI ISL 9627636|2022-01-03
hCoV-19/Iran/Urok-NIC-1454/2021|EPI ISL 9627635|2022-01-04
hCoV-19/Iran/49883/2021|EPI ISL 15755379|2021-07-31
hCoV-19/Iran/Khorasan-NIC-S6/2021|EPI ISL 4803554|2021-08-29
hCoV-19/Iran/AF1236/2021|EPI ISL 15755218|2021-07-03
hCoV-19/Iran/AH1173/2021|EPI ISL 15755180|2021-09-08
hCoV-19/Iran/AM175/2021|EPI ISL 15755548|2021-11-23
hCoV-19/Iran/Ahvaz-NIC-S8/2021|EPI ISL 4803558|2021-08-04
hCoV-19/Iran/AHvaz-NIC-S8/2021|EPI ISL 4803557|2021-08-02
hCoV-19/Iran/NIC-24-1/2021|EPI ISL 8754016|2021-09-04
hCoV-19/Iran/NIC-16-1/2021|EPI ISL 8754009|2021-09-03
hCoV-19/Iran/NIC-US/2021|EPI ISL 8754022|2021-12-15
hCoV-19/Iran/AL222/2021|EPI ISL 15755475|2021-11-09
hCoV-19/Iran/50130/2021|EPI ISL 15755393|2021-08-24
hCoV-19/Iran/AG0524/2021|EPI ISL 15755410|2021-09-15
hCoV-19/Iran/A9946/2021|EPI ISL 15755354|2021-08-03
hCoV-19/Iran/AM173/2021|EPI ISL 15755488|2021-12-06
hCoV-19/Iran/AL2239/2021|EPI ISL 15755424|2021-11-09
hCoV-19/Iran/AK1046NGS/2021|EPI ISL 15755404|2021-09-05
hCoV-19/Iran/50023/2021|EPI ISL 15755385|2021-08-09
hCoV-19/Iran/AM082/2021|EPI ISL 15755547|2021-11-17
hCoV-19/Iran/AL2275/2021|EPI ISL 15755427|2021-11-08
hCoV-19/Iran/AL2210/2021|EPI ISL 15755420|2021-11-11
hCoV-19/Iran/GRC-S1221/2021|EPI ISL 2227204|2021-04-29
hCoV-19/Iran/AG0625-S-Parsa/2021|EPI ISL 5885889|2021

At last, I compared the three optimized trees:

```
# Load necessary packages
library(ape)
```

```
## Warning: package 'ape' was built under R version 4.2.3
```

```
library(phangorn)
```

```
## Warning: package 'phangorn' was built under R version 4.2.3
```

```
# Read in the trees
tree1 <- read.tree("optimized_tree.nwk")
tree2 <- read.tree("optimized_tree_branch.nwk")
tree3 <- read.tree("optimized_tree_model.nwk")

# Calculate pairwise RF distances between all three trees
dist1 <- RF.dist(tree1, tree2)
dist2 <- RF.dist(tree1, tree3)
dist3 <- RF.dist(tree2, tree3)

# Print the pairwise distances
cat("RF distance between optimized_tree.nwk and optimized_tree_branch.nwk:", dist1, "\n")
```

```
## RF distance between optimized_tree.nwk and optimized_tree_branch.nwk: 136
```

```
cat("RF distance between optimized_tree.nwk and optimized_tree_model.nwk:", dist2, "\n")
```

```
## RF distance between optimized_tree.nwk and optimized_tree_model.nwk: 136
```

```
cat("RF distance between optimized_tree_branch.nwk and optimized_tree_model.nwk:", dist3, "\n")
```

```
## RF distance between optimized_tree_branch.nwk and optimized_tree_model.nwk: 0
```

The RF distance is a measure of the topological distance between two trees, and it represents the number of bipartitions that differ between the trees. A distance of 0 indicates that the two trees are identical in their topology, while a higher distance indicates a greater difference in their topology. The output shows that the RF distance between "optimized_tree.nwk" and "optimized_tree_branch.nwk" is 136, indicating a substantial difference in their topology. The RF distance between "optimized_tree.nwk" and "optimized_tree_model.nwk" is also 136, indicating a similar level of difference in topology. However, the RF distance between "optimized_tree_branch.nwk" and "optimized_tree_model.nwk" is 0, indicating that these two trees have identical topologies.

To assess the reliability of the phylogenetic tree, generate multiple bootstrap replicates of the dataset and estimate the phylogenetic tree for each replicate.

Answer:

The bellow code, demonstrates how to generate multiple bootstrap replicates of an alignment file and estimate the phylogenetic tree for each replicate in order to assess the reliability of the resulting tree.

It first reads in an alignment file in PHYLIP format using the read.phyDat() function from the ape package. It then converts the alignment to a DNAbin object using the as.DNAbin() function. Next, the code estimates the phylogenetic tree using the neighbor-joining (NJ) method with the nj() function from the ape package. The code then sets the number of bootstrap replicates to generate (nboot) and uses a for loop to generate the bootstrap replicates and estimate the trees using the NJ method. For each bootstrap replicate, the code saves a plot of the resulting tree as a PNG file using the png(), plot(), and dev.off() functions.

```
# Load the ape package
library(ape)
```

```
## Warning: package 'ape' was built under R version 4.2.3
```

```
library(phangorn)
```

```
## Warning: package 'phangorn' was built under R version 4.2.3
```

```
# Read in your alignment file
aln <- read.phyDat("fereshteh_gisaid_output.fasta.raxml.reduced.phy")

# Convert the alignment to a DNAbin object
dnabin_aln <- as.DNAbin(aln)

# Estimate the phylogenetic tree
tree <- nj(dist.dna(dnabin_aln))

# Set the number of bootstrap replicates
nboot <- 100

# Generate a set of bootstrap replicates and estimate the trees
boot_trees <- vector(mode = "list", length = nboot)
for (i in 1:nboot) {
  boot_dnabin_aln <- as.DNAbin(sample(aln, replace = TRUE))
  boot_trees[[i]] <- nj(dist.dna(boot_dnabin_aln))
}

# Save each bootstrap tree plot as a PNG file
for (i in 1:nboot) {
  # Set the filename for this bootstrap tree plot
  filename <- paste0("boot_tree_", i, ".png")

  # Open a new PNG device and plot the tree
  png(filename, width = 1500, height = 800, res = 100)
  plot(boot_trees[[i]], cex = 0.3)

  # Close the PNG device
  dev.off()
}
```

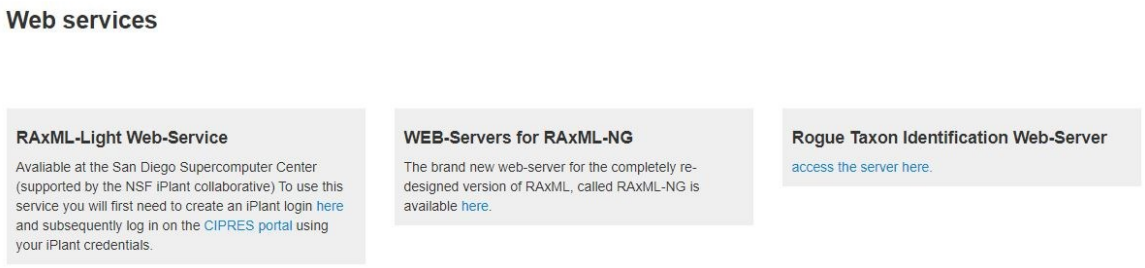
The code generates 100 PNG files and 100 tree files as output, which can be find in the current working directory.

Infer bootstrap support for branches in the optimized tree

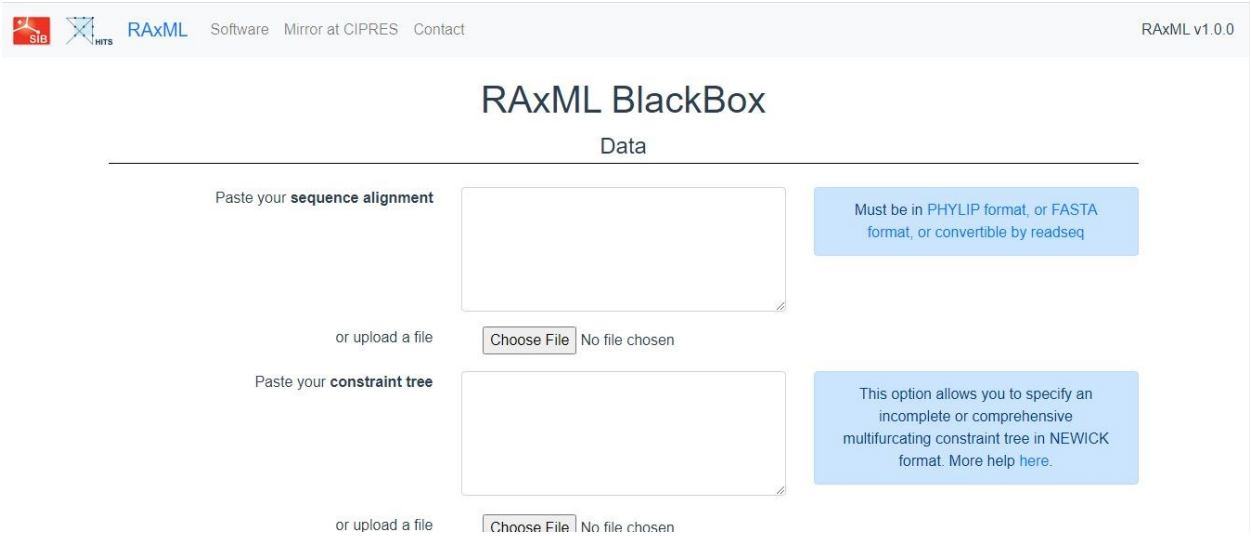
Answer:

The Felsenstein confidence (FC) is a method for calculating bootstrap support values for phylogenetic trees that is based on a resampling technique known as the bootstrap. The basic idea of the FC method is to compare the log-likelihood of the original tree to the log-likelihoods of a set of bootstrap trees, and to calculate a confidence value for each node based on the proportion of bootstrap trees in which the node appears.

RAxML can calculate bootstrap support values for the nodes in the maximum likelihood tree using a resampling technique known as non-parametric bootstrap.



I used RAxML-NG (https://raxml-ng.vital-it.ch/#/).



The output is containing:

sequenceAlignment.phy	7/9/2023 8:56 PM	PHYLIP File	3,687 KB
slurm_raxml.sbatch	7/9/2023 8:56 PM	SBATCH File	1 KB
raxmlArg.txt	7/9/2023 9:27 PM	Text Document	1 KB
result.raxml.bestModel	7/9/2023 9:27 PM	BESTMODEL File	1 KB
result.raxml.bestTree	7/9/2023 9:27 PM	BESTTREE File	10 KB
result.raxml.bestTreeCollapsed	7/9/2023 9:27 PM	BESTTREECOLLAP...	9 KB
result.raxml.log	7/9/2023 9:27 PM	Text Document	16 KB
result.raxml.mlTrees	7/9/2023 9:27 PM	MLTREES File	184 KB
result.raxml.rba	7/9/2023 9:27 PM	RBA File	98 KB
result.raxml.startTree	7/9/2023 9:27 PM	STARTTREE File	184 KB

Data visualization

Use the phylogenetic tree and bootstrap support values to visualize the evolutionary history of the selected strain in Iran. (You may use ggtree package in R)

Answer:

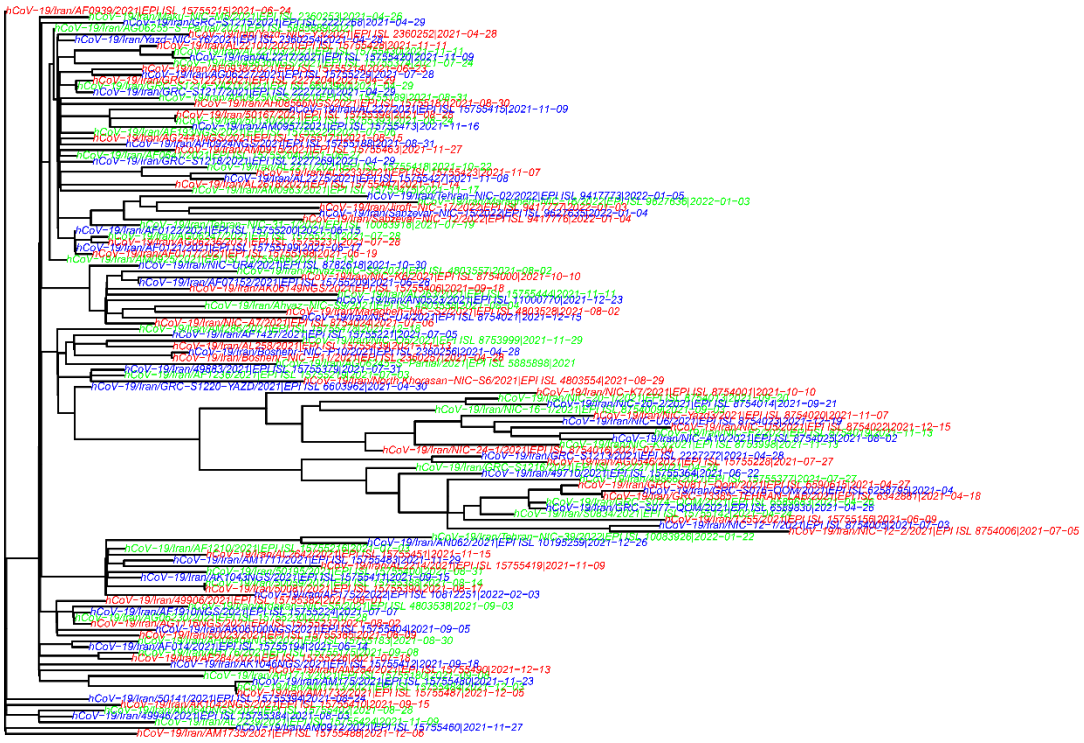
```
# Load necessary packages
library(ape)
```

```
## Warning: package 'ape' was built under R version 4.2.3
```

```
library(phangorn)
```

```
## Warning: package 'phangorn' was built under R version 4.2.3
```

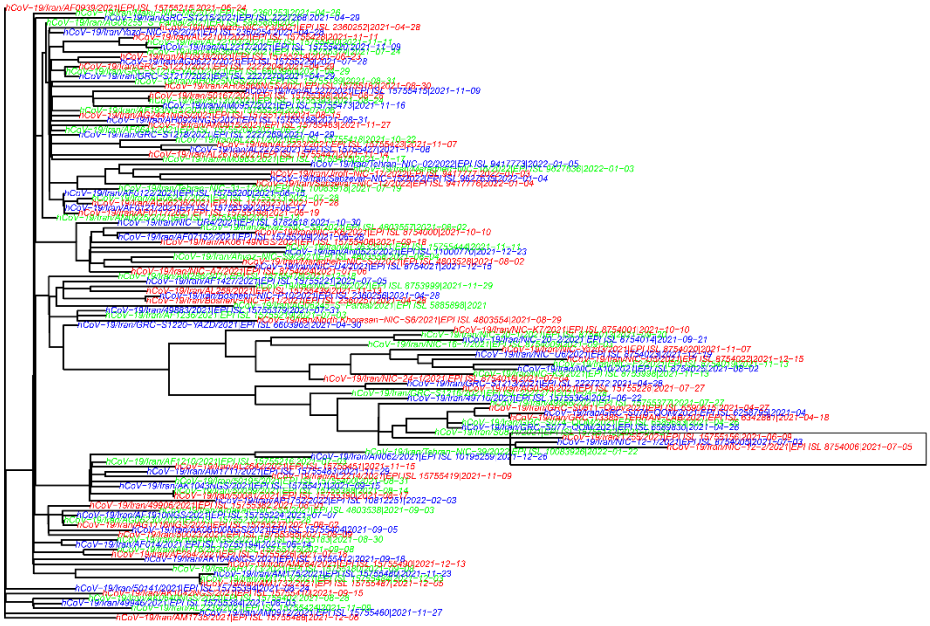
```
# Read in the optimized tree topology
tree_optimized <- read.tree("result.raxml-2.nwk")

# Prune the tree to remove some nodes or branches
tree_pruned <- drop.tip(tree_optimized, c("hCoV-19/Iran/NIC-A7/2021|EPI_ISL_8754024|2021-07-06:1e-08"))

# Collapse multifurcating nodes to simplify the tree
tree_simple <- multi2di(tree_pruned)

# Define a vector of colors for each group
group_colors <- c("red", "blue", "green")

# Assign a color to each tip label based on the group variable
group <- c("Group1", "Group2", "Group1", "Group3", "Group2", "Group3")
tip_colors <- group_colors[as.factor(group)]

# Plot the simplified tree with larger font size, longer branches, and colored tip labels
plot(tree_simple, cex = 0.3, edge.length = 10, tip.color = tip_colors)
```



A rooted phylogenetic tree of the Delta strain of COVID-19 in Iran was generated as the outcome.

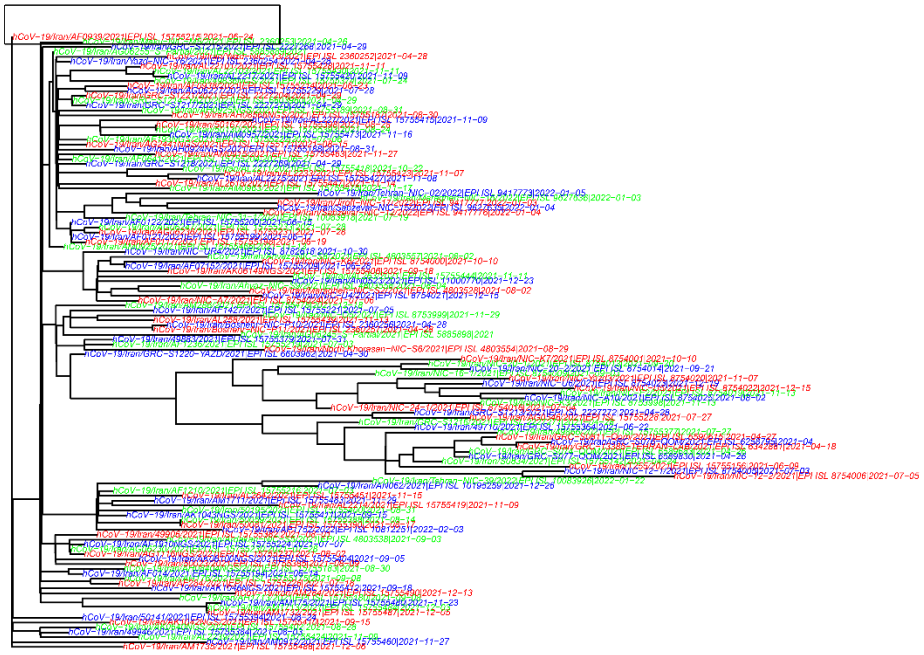
1. The meaning of the longest branch from the root in a phylogenetic tree can depend on the specific context of the analysis. In general, however, a long branch can represent a period of relatively rapid evolution, such as a burst of adaptive radiation or a period of rapid divergence due to environmental or ecological factors. Alternatively, a long branch may simply be the result of a lack of sampling or incomplete information about the evolutionary relationships among the taxa being studied that in my tree is there:

hCoV−19/Iran/NIC−12−2/2021|EPI ISL 8754006|2021−07−05 is the strain that has accumulated a relatively large amount of genetic divergence compared to the other sequences in the tree. This could be due to a variety of factors, such as a longer period of evolution, more mutations or recombination events, or a different evolutionary trajectory compared to the other sequences in the tree.

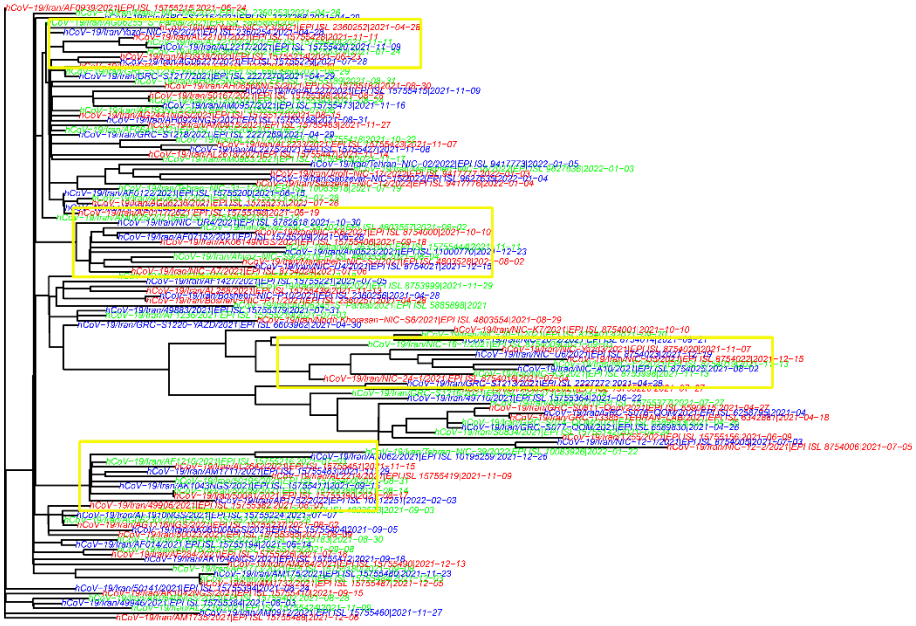


2. The meaning of the shortest branch in a phylogenetic tree can depend on the specific context of the analysis. In general, however, a short branch can indicate recent common ancestor, a period of relatively little evolutionary change, or a lack of genetic diversity among the sequences being analyzed.

hCoV−19/Iran/AF0939/2021|EPI ISL 15755215|2021−06−24 is the strain that is very closely related to the other sequences in the tree and has not diverged much from their common ancestor. This could indicate that this particular sequence is part of a recent outbreak or cluster of cases that has not yet had time to accumulate many genetic changes.



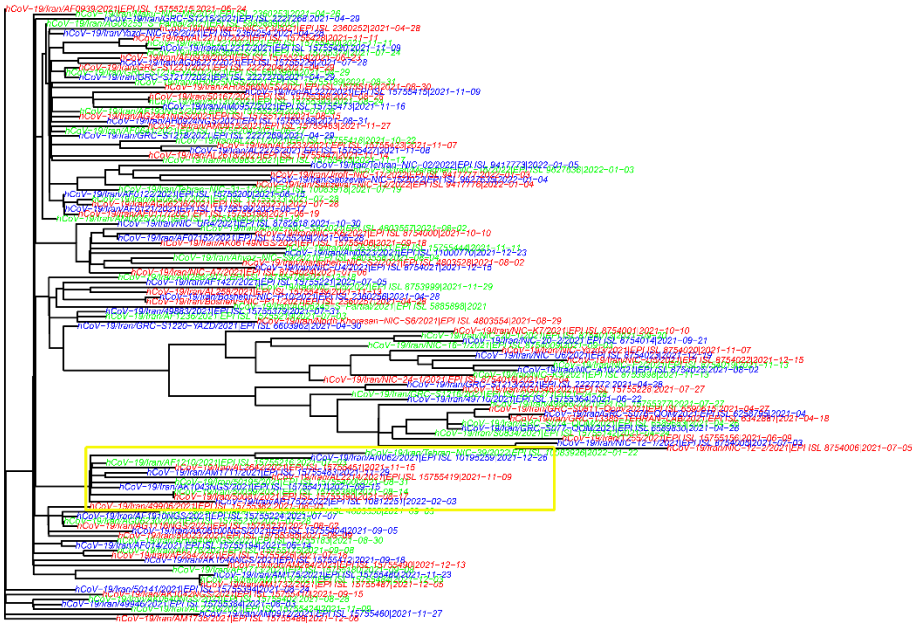
3. In a phylogenetic tree, a clade is a group of organisms that share a common ancestor, and are therefore more closely related to each other than to any other organisms in the tree. In my tree there are some clades that I determined some of them:



4. In my tree some clades that has more taxa in a particular clade in your phylogenetic tree, it could indicate that this group of organisms has undergone more diversification or evolution compared to other clades in the tree. This could be due to factors such as geographic isolation, differences in host range or transmission dynamics, or a combination of genetic and environmental factors that have led to adaptation and divergence over time. Alternatively, the number of taxa in a clade could be influenced by sampling biases in the data, such as differences in the number or quality of sequences available for different regions or time periods. For example, if more sequences have been sampled from a particular region or time period, this could lead to the appearance of a larger clade in the tree.

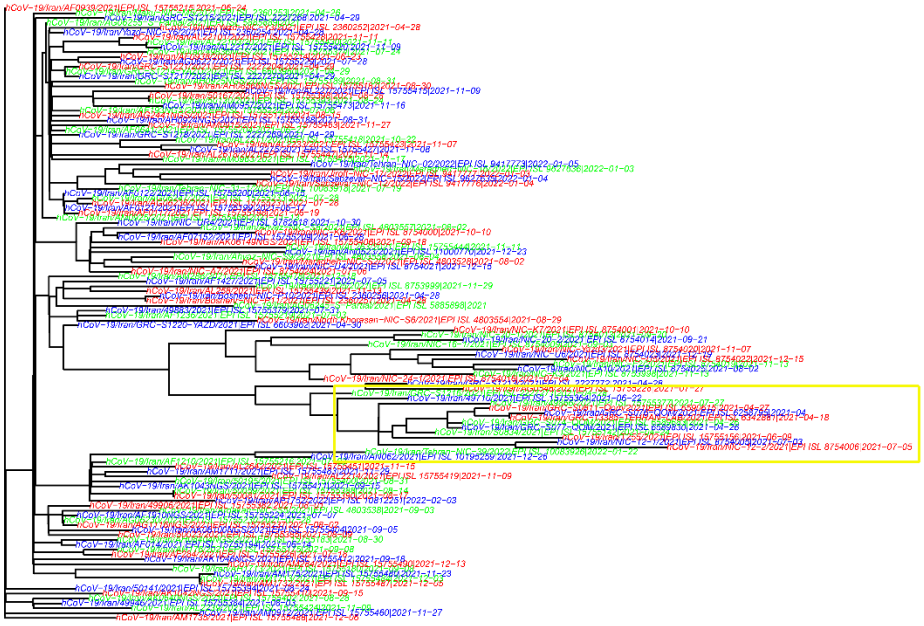
   In my tree there are someclades that are highly concentrated , with closely related taxa clustered closely together in the tree, it may suggest that these organisms have undergone relatively little evolution or divergence from their common ancestor, or that they have diversified rapidly and recently. This could be due to factors such as geographic isolation, host range or transmission dynamics, or the evolution of novel traits or adaptations that allow them to exploit new ecological niches.

For example:



5. Also in my tree there are some clade that are more dispersed, with closely related taxa scattered across different parts of the tree, it may suggest that these organisms have undergone a greater degree of evolutionary change or diversification. This could be due to factors such as longer periods of time since their common ancestor, more complex evolutionary histories involving multiple speciation events or hybridization events, or more gradual patterns of adaptation and diversification.

For example:



Therefore, in conclusion, my phylogenetic tree of the Delta strain of COVID-19 in Iran displays a diverse range of strains and clades that can provide valuable insights into the evolutionary relationships and transmission patterns of the virus in the region. The longest branch in the tree may indicate a period of rapid evolution due to environmental or ecological factors, while the shortest branch may suggest a recent common ancestor, indicating that the sequence is part of a recent outbreak or cluster of cases with few genetic changes.

The number of taxa in a clade can indicate the level of diversification or evolution compared to other clades in the tree, due to factors such as geographic isolation, differences in host range or transmission dynamics, or genetic and environmental factors. Highly concentrated clades with closely related taxa clustered closely together in the tree may suggest that these organisms have

undergone little evolution or divergence from their common ancestor, or that they have diversified rapidly and recently due to novel traits or adaptations that allow them to exploit new ecological niches.