Department of Statistics Savitribai Phule Pune University.

ST-O19: Statistical Methods for Bio-computing On Mumps Jan- May 2023

On the topic

Mumps

Guided by: Prof. M. M. Kale.

Submitted by:

Mohit Jadhav (2118)
Pavan Kaulapure (2127)

Dated:

15th April 2023

INTRODUCTION TO MUMPS:

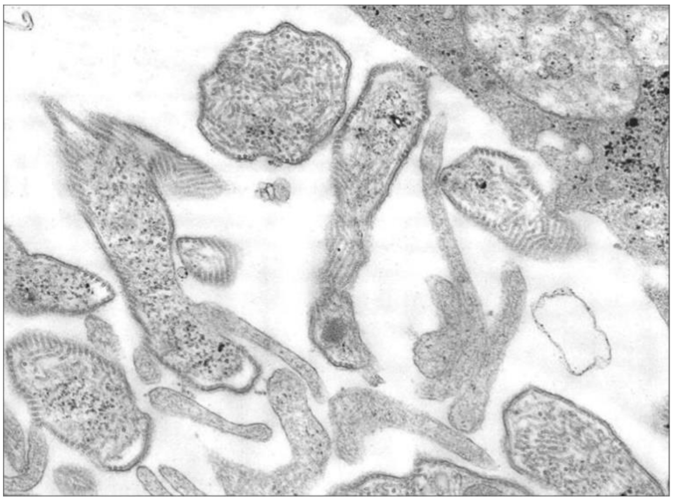


Fig1. Transmission electron micrograph showing the ultra-structural details of mumps virions grown in Vero cells.

Mumps is best known as a common childhood viral disease and is characterized by swelling of the parotid gland. Mumps virus, the causative agent of mumps infection, is an enveloped RNA virus that belongs to the genus Rubulavirus in the family Paramyxoviridae. In electron microscopy, the virion presents as a particle with a shape that varies between spherical and pleiomorphic with a diameter of about 200 nm.

The viral genome is contained in a linear molecule of single-stranded, negative-strand RNA, 15 384 nucleotides in length, which encodes six structural proteins and at least two non-structural proteins. The capsid consists of the major structural nucleocapsid protein, the phosphoprotein, and the large protein; the last two are thought to constitute RNA polymerase. The envelope is a lipid bilayer membrane composed of the matrix protein and two surface glycoproteins. The surface glycoproteins—haemagglutinin-neuraminidase and fusion protein—bring about viral adsorption and fusion of the virion membrane with the host cell membrane, respectively; both are needed for cell-to-cell fusion. Virion membrane fusion seems to be associated with neurovirulence. The lipid membrane renders the virus susceptible to ether and alcoholic disinfectants. The virus is stable at 4°C for days. Population genetics of mumps have been based on genotyping of the small hydrophobic gene, the most variable part of the viral genome. The function of the protein it encodes is not known. Genotypes show nucleotide variation of 2–4% within genotypes and at least 6% between genotypes. 12 mumps virus genotypes, designated A to L, have been described and their geographic distribution varies: in the western hemisphere, genotypes C, D, E, G, and H prevail, and in Asian countries, genotypes B, F, and I predominate. Several genotypes might circulate simultaneously in a region, and there can be temporal shifts in genotype distribution the factors that drive genotype distribution are not known. Mumps virus is not classified into serotypes; however, findings in vivo and in vitro suggest that cross-neutralization between genotypes might be reduced. The significance and effect of reduced cross-neutralization between genotypes with respect to mumps epidemiology and vaccination remain to be established.

DATA DESCRIPTION

From the given database of sequences, we have chosen ten sequences of the Hepatitis B virus with the following accession numbers:

1. **AD90231**
2. **AD00663**
3. **CX63709**
4. **CY08214**
5. **DAF142766**
6. **DAF142769**
7. **FZ77158**
8. **FZ77160**
9. **KAF365891**
10. **MEU069917**

These 10 sequences are DNA sequences consisting of 4 nucleotides:

Adenine (A), Guanine (G), Cytosine (C), Thymine (T) etc.

SOFTWARES AND PACKAGES USED:

Software used: R-Studio, Excel

Packages used:

seqinr

entropy

infotheo

phangorn

ape

markovchain

Question. 1)

Compute entropy for each sequence. Also compute mutual information content between every pair of sequences by taking

- **First 10% terms.**
- **Middle 10% terms.**
- **Last 10% terms.**
- **Complete sequence.**

Adjust this proportion to equal length by approximately adding or removing some terms.

Comment on the result. Store the result in appropriate format.

Solution:

Adjusting proportion to equal length:

We have to find the entropy for each sequence and also mutual information content between every pair of sequences. For taking the first 10% terms, middle 10% terms and last 10% terms, we need sequences having the same length. We have chosen 10 DNA sequences from MUMPS. Out of 10, 9 sequences have length 318 while only one sequence i.e. sequence no. KAF365891 have length 271. That's why we need some adjustment for proportions and the simplest way is by taking minimum length of all the sequences. So, the minimum length is 271. Hence, we take all the sequences up to length 271 only. In this way without adding or removing any of the nucleotides, we have adjusted proportions to equal length.

Now,

10% of 271 is 27.1 ≈ 27.

Hence the first 10% terms are from 1 to 322.

Middle 10 % terms are from 124 to 150.

The last 10% terms are from 245 to 271.

Entropy:

Entropy is a measure of average uncertainty of an outcome. The entropy of a variable is the "amount of information" contained in the variable.

Let X be a random variable having values x_1, x_2, x_3,…….,x_m with probabilities p_1,p_2,p_3,…….p_m. The entropy also known as Shannon's entropy of X is given by,

$$H(X) = -\sum_{i} pi\ log2(pi)$$

The range of entropy is 0 ≤ Entropy ≤ log(n) where n is the length of sequences.

In biological sequence analysis, higher the entropy value, higher is the uncertainty in the appearance of nucleotides at a particular way which means that sequence/organism is highly evolving. On the other hand, low entropy value is an indicator of sequence/organism is somewhat conserved. Generally, entropy is measured in “bits”. Maximum entropy a DNA sequence can have is “2 bits”.

Now,

Aim: To compute entropy for each sequence.

```
> rm(list=ls())

> library('seqinr')
> D=read.fasta(file.choose(),seqtype = "DNA")

> s1=D$AD90231[1:271]
> s2=D$AD00663[1:271]
> s3=D$CX63709[1:271]
> s4=D$CY08214[1:271]
> s5=D$DAF142766[1:271]
> s6=D$DAF142769[1:271]
> s7=D$FZ77158[1:271]
> s8=D$FZ77160[1:271]
> s9=D$KAF365891[1:271]
> s10=D$MEU069917[1:271]

> S=list(s1,s2,s3,s4,s5,s6,s7,s8,s9,s10)




> ###gsub() function in R Language is used to replace all
the matches of

> #a pattern from a string
> library(entropy)
> H_using_package=c()
> New=list()
> for(i in 1:10)
+ {
+   a=gsub("a","1",S[[i]])
+   g=gsub("g",2,a)
+   c=gsub("c","3",g)
+   New[[i]]=(gsub("t","4",c))
+   H_using_package[i]=entropy.empirical(table(New[[i]
]),"log2")
+ }


> H_using_package
[1] 1.981608 1.981902 1.987290 1.983416
[5] 1.975771 1.976761 1.981467 1.987666
[9] 1.978193 1.969965



> ##Entropy by using formula:
> n=length(S[[1]])
> A=c();G=c();C=c();T=c()
> p=c()
> H=c()
> for(i in 1:10)
+ {
+   A[i]=length(subset(S[[i]],S[[i]]=="a"))
+   G[i]=length(subset(S[[i]],S[[i]]=="g"))
+   C[i]=length(subset(S[[i]],S[[i]]=="c"))
+   T[i]=length(subset(S[[i]],S[[i]]=="t"))
+   p[[i]]=c(A[i]/n,G[i]/n,C[i]/n,T[i]/n)
+   H[i]=sum((-p[[i]])*log2(p[[i]]))
+ }
> H
[1] 1.981608 1.981902 1.987290 1.983416
[5] 1.975771 1.976761 1.981467 1.987666
[9] 1.978193 1.969965



> table1=data.frame(H,H_using_package)
> table1
H H_using_package
1  1.981608       1.981608
2  1.981902       1.981902
3  1.987290       1.987290
4  1.983416       1.983416
5  1.975771       1.975771
6  1.976761       1.976761
7  1.981467       1.981467
8  1.987666       1.987666
9  1.978193       1.978193
10 1.969965        1.969965
```

The following table gives entropy for the each of the sequence.

Sequence Accession No.	Entropy by using formula	Entropy by package entropy	Entropy wise ranking
AD90231	1.98168	1.981608	6
AD00663	1.98192	1.981902	7
CX63709	1.98720	1.987290	9
CY08214	1.983416	1.983416	8
DAF142766	1.97571	1.975771	2
DAF142769	1.97671	1.976761	3
FZ77158	1.98147	1.981467	5
KAF365891	1.987666	1.987666	10
FZ77160	1.97813	1.978193	4
MEU06997	1.969965	1.969965	1

Interpretation:

Entropy obtained by using formula and using package entropy are exactly the same.

The entropy values for all ten sequences are nearly equal. That means amount of uncertainty and complexity from all the sequences is same. Entropy refers to uncertainty. Hence, all the 10 sequences contain almost equal amount of randomness andvariation.

Mutual Information Content:

Mutual Information Content can be defined as-

$I(x)=H_{before} - H_{after}$

I.e. $I(x)=Entropy_{before} - Entropy_{after}$

Mutual information is one of many quantities that measures how much one random variable tells us about another. It is a measure of the mutual dependence between the two variables. It is a dimensionless quantity with (generally) units of bits, and can be thought of as the reduction in uncertainty about one random variable given knowledge of another. It quantifies the ‘amount of information’ obtained about one sequence, through the other sequence.

- High mutual information indicates a large reduction in the amount of uncertainty and hence a high level of association.
- low mutual information indicates a small reduction and hence weak level of association.
- zero mutual information between two random variables means there is no association.
- Mutual information content is symmetric. i.e. $M(X, Y) = M(Y, X)$

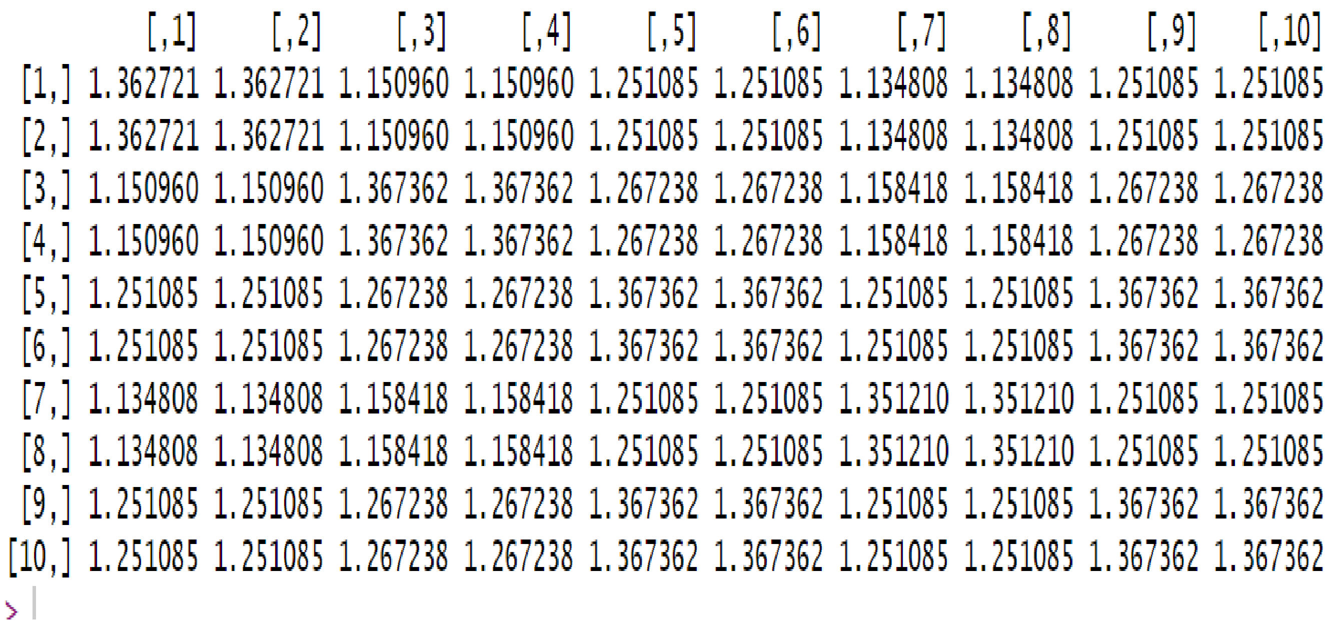
Aim: To compute mutual information content in every pair of the sequences.

```
> c=combn(1:10,2)
```

We want to compute mutual information contained in every pair of sequences. So, we havetotal 45 pairs of sequences.

## M.I.C. for first 10 % terms:

```
> ##MIC for first 10% terms:
> #install.packages('infotheo')
> library('infotheo')
> m=matrix(nrow=10,ncol=27)
> for(i in 1:10)
+ {
+   m[i,]=S[[i]][1:27]
+ }
> MIC1=matrix(nrow=10,ncol=10)
> for(i in 1:10)
+ {
+   for(j in 1:10)
+    {
+      MIC1[i,j]=mutinformation(m[i,],m[j,])
+   }
+ }
> round(MIC1,6)
```



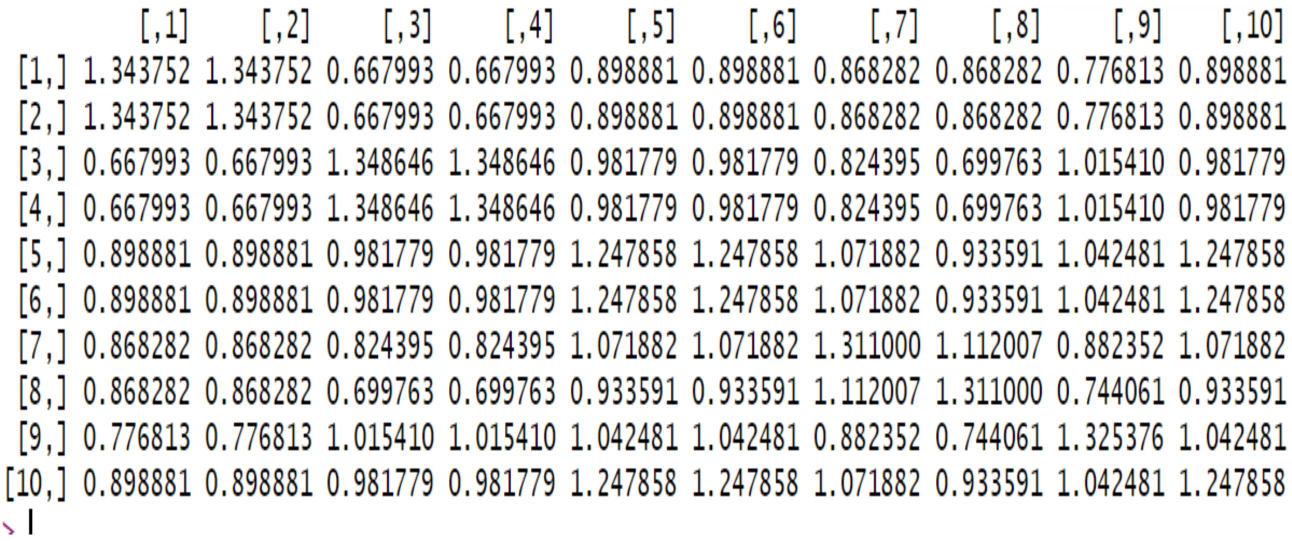
```
         [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]     [,9]    [,10]
 [1,] 1.362721 1.362721 1.150960 1.150960 1.251085 1.251085 1.134808 1.134808 1.251085 1.251085
 [2,] 1.362721 1.362721 1.150960 1.150960 1.251085 1.251085 1.134808 1.134808 1.251085 1.251085
 [3,] 1.150960 1.150960 1.367362 1.367362 1.267238 1.267238 1.158418 1.158418 1.267238 1.267238
 [4,] 1.150960 1.150960 1.367362 1.367362 1.267238 1.267238 1.158418 1.158418 1.267238 1.267238
 [5,] 1.251085 1.251085 1.267238 1.267238 1.367362 1.367362 1.251085 1.251085 1.367362 1.367362
 [6,] 1.251085 1.251085 1.267238 1.267238 1.367362 1.367362 1.251085 1.251085 1.367362 1.367362
 [7,] 1.134808 1.134808 1.158418 1.158418 1.251085 1.251085 1.351210 1.351210 1.251085 1.251085
 [8,] 1.134808 1.134808 1.158418 1.158418 1.251085 1.251085 1.351210 1.351210 1.251085 1.251085
 [9,] 1.251085 1.251085 1.267238 1.267238 1.367362 1.367362 1.251085 1.251085 1.367362 1.367362
[10,] 1.251085 1.251085 1.267238 1.267238 1.367362 1.367362 1.251085 1.251085 1.367362 1.367362
> 
```

Interpretations:

None of mutual information between two sequences is exactly zero. Hence, every pair of sequences have some association.
Mutual information between almost every pair of sequences indicate high level of association.
Entropies are approximately same because first terms of all sequences are almost same.

## M.I.C. for middle 10 % terms:

```
> ####MIC for middle 10% terms
> m=matrix(nrow=10,ncol=27)
> for(i in 1:10)
+ {
+  m[i,]=S[[i]][124:150]
+ }
> MIC2=matrix(nrow=10,ncol=10)
> for(i in 1:10)
+ {
+  for(j in 1:10)
+   {
+     MIC2[i,j]=mutinformation(m[i,],m[j,])
+   }
+ }
> round(MIC2,6)
```



```
         [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]     [,9]    [,10]
 [1,] 1.343752 1.343752 0.667993 0.667993 0.898881 0.898881 0.868282 0.868282 0.776813 0.898881
 [2,] 1.343752 1.343752 0.667993 0.667993 0.898881 0.898881 0.868282 0.868282 0.776813 0.898881
 [3,] 0.667993 0.667993 1.348646 1.348646 0.981779 0.981779 0.824395 0.699763 1.015410 0.981779
 [4,] 0.667993 0.667993 1.348646 1.348646 0.981779 0.981779 0.824395 0.699763 1.015410 0.981779
 [5,] 0.898881 0.898881 0.981779 0.981779 1.247858 1.247858 1.071882 0.933591 1.042481 1.247858
 [6,] 0.898881 0.898881 0.981779 0.981779 1.247858 1.247858 1.071882 0.933591 1.042481 1.247858
 [7,] 0.868282 0.868282 0.824395 0.824395 1.071882 1.071882 1.311000 1.112007 0.882352 1.071882
 [8,] 0.868282 0.868282 0.699763 0.699763 0.933591 0.933591 1.112007 1.311000 0.744061 0.933591
 [9,] 0.776813 0.776813 1.015410 1.015410 1.042481 1.042481 0.882352 0.744061 1.325376 1.042481
[10,] 0.898881 0.898881 0.981779 0.981779 1.247858 1.247858 1.071882 0.933591 1.042481 1.247858
> 
```

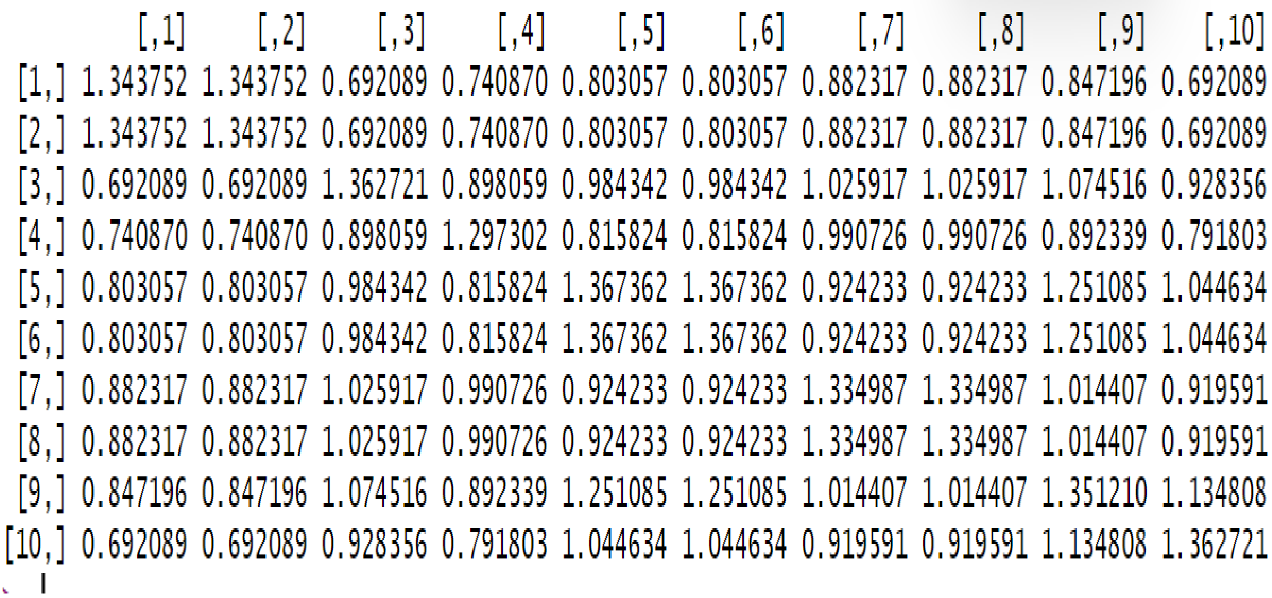
Interpretations:

i) **None of mutual information between two sequences is exactly zero. Hence, every pair have sequence have some association.**

ii) **Mutual information between every sequence pair indicate comparatively moderate level of association.**

## M.I.C. for last 10 % terms:

```
> m=matrix(nrow=10,ncol=27)
> for(i in 1:10)
+ {
+  m[i,]=S[[i]][245:271]
+ }
> MIC3=matrix(nrow=10,ncol=10)
> for(i in 1:10)
+ {
+  for(j in 1:10)
+   {
+     MIC3[i,j]=mutinformation(m[i,],m[j,])
+   }
+ }
> round(MIC3,6)
```



```
          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]     [,9]    [,10]
 [1,] 1.343752 1.343752 0.692089 0.740870 0.803057 0.803057 0.882317 0.882317 0.847196 0.692089
 [2,] 1.343752 1.343752 0.692089 0.740870 0.803057 0.803057 0.882317 0.882317 0.847196 0.692089
 [3,] 0.692089 0.692089 1.362721 0.898059 0.984342 0.984342 1.025917 1.025917 1.074516 0.928356
 [4,] 0.740870 0.740870 0.898059 1.297302 0.815824 0.815824 0.990726 0.990726 0.892339 0.791803
 [5,] 0.803057 0.803057 0.984342 0.815824 1.367362 1.367362 0.924233 0.924233 1.251085 1.044634
 [6,] 0.803057 0.803057 0.984342 0.815824 1.367362 1.367362 0.924233 0.924233 1.251085 1.044634
 [7,] 0.882317 0.882317 1.025917 0.990726 0.924233 0.924233 1.334987 1.334987 1.014407 0.919591
 [8,] 0.882317 0.882317 1.025917 0.990726 0.924233 0.924233 1.334987 1.334987 1.014407 0.919591
 [9,] 0.847196 0.847196 1.074516 0.892339 1.251085 1.251085 1.014407 1.014407 1.351210 1.134808
[10,] 0.692089 0.692089 0.928356 0.791803 1.044634 1.044634 0.919591 0.919591 1.134808 1.362721
```

Interpretations:

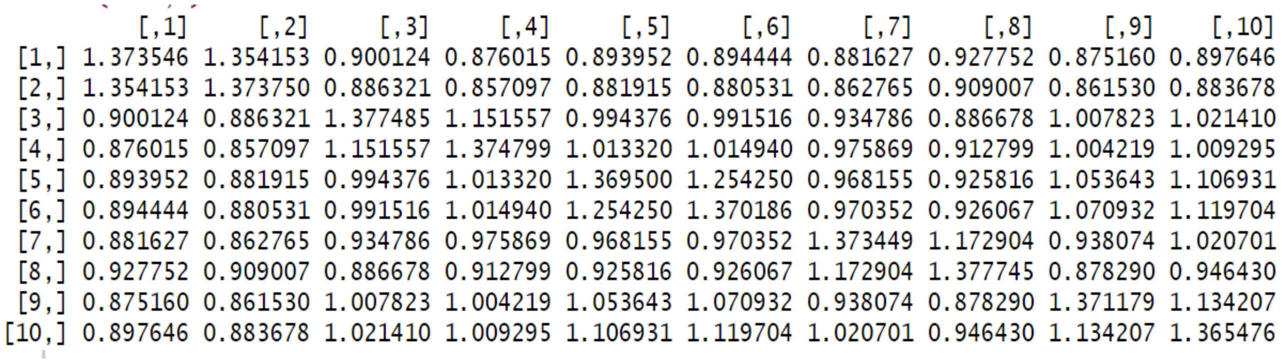
i) None of mutual information between two sequences is exactly zero. Hence, every pair of sequences has some association.

ii) Mutual information between sequence 5 and 6 is high indicating very high level of association.

iii) In last 10% terms, mutual information is different indicating sequences are random at the end.

## M.I.C. for complete sequence:

```
> ###MIC for complete sequence
> m=matrix(nrow=10,ncol=271)
> for(i in 1:10)
+ {
+  m[i,]=S[[i]][1:271]
+ }
> MIC4=matrix(nrow=10,ncol=10)
> for(i in 1:10)
+ {
+   for(j in 1:10)
+    {
+     MIC4[i,j]=mutinformation(m[i,],m[j,])
+    }
+ }
> round(MIC4,6)
```



```
          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]     [,9]    [,10]
 [1,] 1.373546 1.354153 0.900124 0.876015 0.893952 0.894444 0.881627 0.927752 0.875160 0.897646
 [2,] 1.354153 1.373750 0.886321 0.857097 0.881915 0.880531 0.862765 0.909007 0.861530 0.883678
 [3,] 0.900124 0.886321 1.377485 1.151557 0.994376 0.991516 0.934786 0.886678 1.007823 1.021410
 [4,] 0.876015 0.857097 1.151557 1.374799 1.013320 1.014940 0.975869 0.912799 1.004219 1.009295
 [5,] 0.893952 0.881915 0.994376 1.013320 1.369500 1.254250 0.968155 0.925816 1.053643 1.106931
 [6,] 0.894444 0.880531 0.991516 1.014940 1.254250 1.370186 0.970352 0.926067 1.070932 1.119704
 [7,] 0.881627 0.862765 0.934786 0.975869 0.968155 0.970352 1.373449 1.172904 0.938074 1.020701
 [8,] 0.927752 0.909007 0.886678 0.912799 0.925816 0.926067 1.172904 1.377745 0.878290 0.946430
 [9,] 0.875160 0.861530 1.007823 1.004219 1.053643 1.070932 0.938074 0.878290 1.371179 1.134207
[10,] 0.897646 0.883678 1.021410 1.009295 1.106931 1.119704 1.020701 0.946430 1.134207 1.365476
```

Interpretations:

i) **None of mutual information between two sequences is exactly zero. Hence, every pair of sequences has some association.**

ii) **Mutual information between sequence 5 and 6 is high indicating very high level of association.**

Question 2)

Using UPGMA algorithm reconstruct a phylogenetic tree topology for this group of sequences with distance function as -

- **Difference between entropy of two sequences.**
- **Frequency of A, G, C and T based distance function of your choice.**
- **Any distance function you have chosen. Comment on results.**

Solution:

Terminologies used:

- **Operational taxonomic units (OTUs):** The known nodes in the phylogenetic tree
- **Phylogenetic tree:** A two-dimensional graph depicting nodes and branches that illustrates evolutionary relationships between molecules and organisms using sequences
- **Nodes:** The points that connect branches and usually represent the taxonomic units
- **Branches:** A branch connects any two nodes

The phylogeny reconstruction method results in phylogenetic tree which may or may notcorroborate with the true phylogenetic tree. There are various methods of phylogeny reconstruction that are divided into two major groups:

The phylogeny reconstruction method results in phylogenetic tree which may or may notcorroborate with the true phylogenetic tree. There are various methods of phylogeny reconstruction that are divided into two major groups:

1). Character based

- Maximum Parsimony (MP)
- Maximum Likelihood (ML)

2). Distance based

- Neighbour-Joining (N-J)
- Un-weighted Pair Group Method with Arithmetic Mean (UPGMA)

UPGMA:

UPGMA is nothing but Unweighted Pair-Group Method with Arithmetic means.
This is the simplest distance-based method of tree construction.

It was originally developed for constructing taxonomic phenograms, i.e. trees that reflect the phenotypic similarities between OTUs (operational taxonomic unit), but it can also be used to construct phylogenetic trees if the rates of evolution are approximately constant among the different lineages.

First identify among all the OTUs, the two OTUs that are most similar to each other and then treat these as a new single OTU. Such OTU is referred to as a composite OTU. Subsequently among the new group of OTUs, identify the pair with the highest similarity, and so on, until left with only two OTUs.

Assumption:
The rate of evolution is approximately constant among different lineages so that an approximate linear relationship exists between evolutionary distance and divergence time.

Principle of working: Principle of decreasing similarity. The most similar sequences willbe clustered first then next best similar and so on.

Algorithm:

Step 1) Decide the distance function in an optimal way.

Step 2) Initialization: dij=0 for all i,j

Step 3) Calculate the pairwise distance using a distance function chosen.

Step 4) Arrange the pairwise distance function into a matrix having diagonal entries as '0'.

Step 5) Choose a pair of distance from a collection of distance values (distance matrix) di*j* such that min dij = di*j* where i != j

Step 6) Connect the sequence i* and j* by a branch having length di*j*/2 to an ancestor**.**

Step 7) Recalculate distance matrix D1=((dij*)) by the formula,

di,(j,m) = (dij+dim)/2

d(i,j),(m,k) = (dim+dik+djm+djk)/4

di,(j,k,l) = (dij+dik+dil)/3

and branch lengths,

li,(j,m) = di,(j,m)/2

l(i,j)(m,k) = d(i,j)(m,k)/2

li,(j,k,l) = di,(j,k,l)/2

Step 8) If there are sequences left in the database go to step 5 with modified distance matrix.

Step 9) Print the tree structure and stop.

- **Molecular Clock Property:**

  UPGMA always produces rooted tree. The time of divergence of any organism is simply the sum of the branch length which are the part of path leading to an organism from an ancestor. Such a property is called as molecular clock property.

- **Verification** of Molecular Clock Property is done by using ultrametric condition:
  - **Ultrametric condition:**

The distances dij are said to be ultrametric if for every triplet of sequences xi, xj, xk, the distances dij, dik, djk either all are equal or two are equal and remaining one is smaller. Satisfaction of ultrametric condition implies constant evolution rate.

- **Four-point condition:**

In general, if a distance matrix is to be represented faithfully by a tree, it must satisfy the following four-point condition,

$$d(i,j) + d(m,n) \leq \max \{d(i,m) + d(j,n) , d(i,n) + d(j,m)\}$$

This requirement implies that typical biological trees will not uniquely represent a given biological distance matrix.

A matrix that satisfies the four-point condition is called additive.

Aim: To construct UPGMA phylogenetic tree topology for this ten group of sequences by using distance function as-

i) Difference between entropy of two sequences

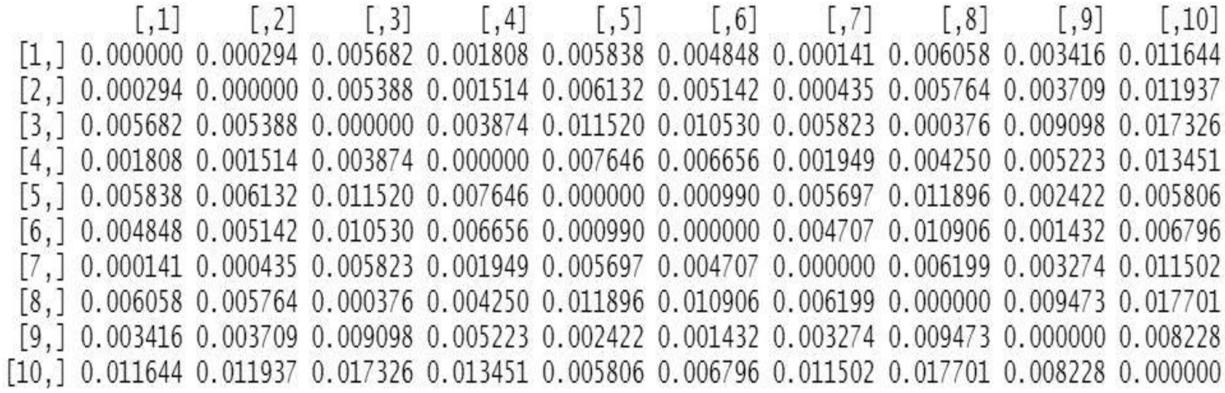
ii) Frequency of A, G, C, T based distance

iii) sup norm distance

i) Constructing UPGMA phylogenic tree with distance function as difference between entropy of two sequences:

```
> d1=matrix(0,nrow=10,ncol=10)
> for(i in 1:10)
+ {
+     for(j in 1:10)
+     {
+         d1[i,j]=abs(H[i]-H[j])
```

```
  +      }
+ }

  > round (d1,6)
```

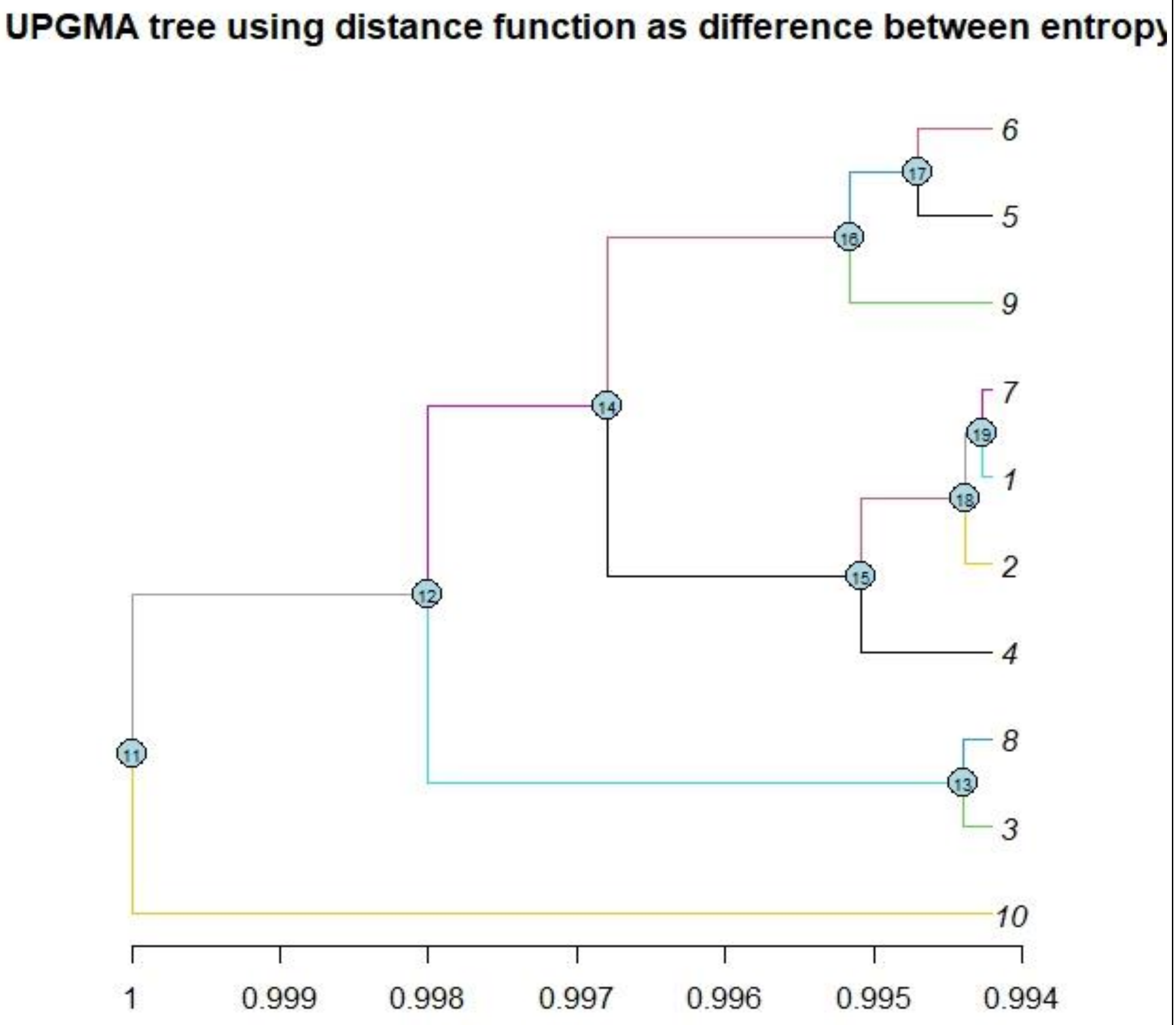


```
          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]     [,9]    [,10]
 [1,] 0.000000 0.000294 0.005682 0.001808 0.005838 0.004848 0.000141 0.006058 0.003416 0.011644
 [2,] 0.000294 0.000000 0.005388 0.001514 0.006132 0.005142 0.000435 0.005764 0.003709 0.011937
 [3,] 0.005682 0.005388 0.000000 0.003874 0.011520 0.010530 0.005823 0.000376 0.009098 0.017326
 [4,] 0.001808 0.001514 0.003874 0.000000 0.007646 0.006656 0.001949 0.004250 0.005223 0.013451
 [5,] 0.005838 0.006132 0.011520 0.007646 0.000000 0.000990 0.005697 0.011896 0.002422 0.005806
 [6,] 0.004848 0.005142 0.010530 0.006656 0.000990 0.000000 0.004707 0.010906 0.001432 0.006796
 [7,] 0.000141 0.000435 0.005823 0.001949 0.005697 0.004707 0.000000 0.006199 0.003274 0.011502
 [8,] 0.006058 0.005764 0.000376 0.004250 0.011896 0.010906 0.006199 0.000000 0.009473 0.017701
 [9,] 0.003416 0.003709 0.009098 0.005223 0.002422 0.001432 0.003274 0.009473 0.000000 0.008228
[10,] 0.011644 0.011937 0.017326 0.013451 0.005806 0.006796 0.011502 0.017701 0.008228 0.000000
```

This is the distance matrix obtained using distance function as difference between entropyof two sequences.

UPGMA method:

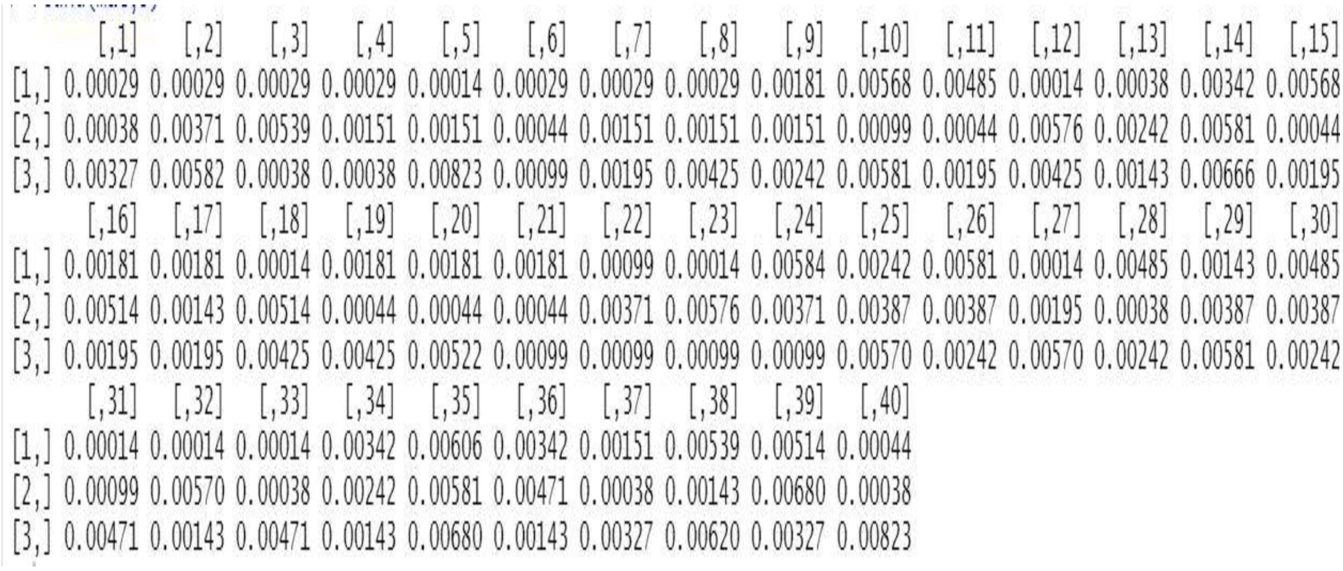
```
> library(phangorn)

> library(ape)

> tree=upgma(d1,method="average")
>row.names(d1)=c("AD90231","AD00663","CX63709","CY08214","DAF142766","DAF142769","FZ77158","FZ77160","KAF365891","MEU069917")
> branch_name1=round(branching.times(tree),5)
>branch_name1
     11      12      13      14      15      16      17      18      19
0.00580 0.00380 0.00019 0.00260 0.00088 0.00096 0.00050 0.00018 0.00007

> plot(tree,main="UPGMA tree using distance function as difference between entropy",type="phylogram",adj=0.5,egde.width=2,edge.color=1:10)
> nodelabels(tree$node.label,cex=0.6,frame="circle")
> axisPhylo(side=1,root.time=TRUE)
```



UPGMA tree using distance function as difference between entropy

## Ultrametric condition:

```
> D = as.matrix(d1)

> w = combn(1:10,3)

> w1 = matrix(0,3,120)
> nw = matrix(0,2,120)
> mini = c()
> ultcon1 = c()
> for(i in 1:120)
+ {
+   w1[1,i] = D[w[1,i],w[2,i]]
+   w1[2,i] = D[w[1,i],w[3,i]]
+  w1[3,i] = D[w[2,i],w[3,i]]
+  mini[i] = which.min(w1[,i])
+  nw[,i] = w1[-mini[i],i]
+  ultcon1[i] = abs(nw[1,i]-nw[2,i])
+  r=round(ultcon1,6)
+ }
> mat = matrix(r,nrow=3,byrow=TRUE)
> round(mat,5)
```



```
        [,1]    [,2]    [,3]    [,4]    [,5]    [,6]    [,7]    [,8]    [,9]   [,10]   [,11]   [,12]   [,13]   [,14]   [,15]
[1,] 0.00029 0.00029 0.00029 0.00029 0.00014 0.00029 0.00029 0.00029 0.00181 0.00568 0.00485 0.00014 0.00038 0.00342 0.00568
[2,] 0.00038 0.00371 0.00539 0.00151 0.00151 0.00044 0.00151 0.00151 0.00151 0.00099 0.00044 0.00576 0.00242 0.00581 0.00044
[3,] 0.00327 0.00582 0.00038 0.00038 0.00823 0.00099 0.00195 0.00425 0.00242 0.00581 0.00195 0.00425 0.00143 0.00666 0.00195
       [,16]   [,17]   [,18]   [,19]   [,20]   [,21]   [,22]   [,23]   [,24]   [,25]   [,26]   [,27]   [,28]   [,29]   [,30]
[1,] 0.00181 0.00181 0.00014 0.00181 0.00181 0.00181 0.00099 0.00014 0.00584 0.00242 0.00581 0.00014 0.00485 0.00143 0.00485
[2,] 0.00514 0.00143 0.00514 0.00044 0.00044 0.00044 0.00371 0.00576 0.00371 0.00387 0.00387 0.00195 0.00038 0.00387 0.00387
[3,] 0.00195 0.00195 0.00425 0.00425 0.00522 0.00099 0.00099 0.00099 0.00099 0.00570 0.00242 0.00570 0.00242 0.00581 0.00242
       [,31]   [,32]   [,33]   [,34]   [,35]   [,36]   [,37]   [,38]   [,39]   [,40]
[1,] 0.00014 0.00014 0.00014 0.00342 0.00606 0.00342 0.00151 0.00539 0.00514 0.00044
[2,] 0.00099 0.00570 0.00038 0.00242 0.00581 0.00471 0.00038 0.00143 0.00680 0.00038
[3,] 0.00471 0.00143 0.00471 0.00143 0.00680 0.00143 0.00327 0.00620 0.00327 0.00823
```

This is the output of the ultrametric condition.

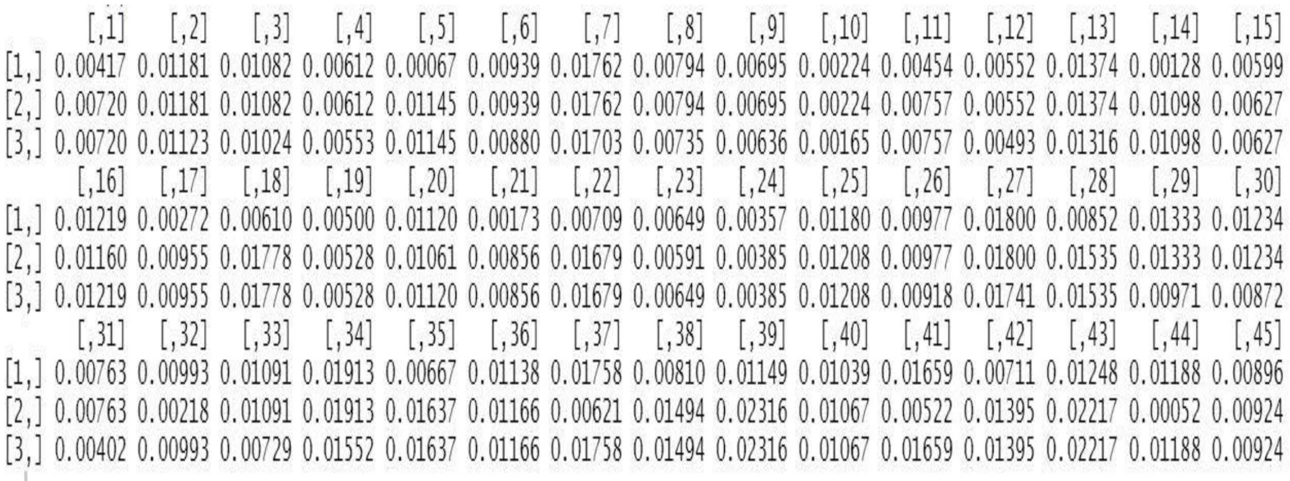
Interpretation:

Neither any of the distances among a triplet is equal nor two are equal and third is smaller than that. So, the ultrametric condition is not satisfied.

Hence, the phylogenetic tree topology using distance function as difference between entropy of two sequences cannot be said to be reliable.

## Four Point Condition:

```
> com=combn(1:10,4)

> dp=matrix(0,3,45)

> for(i in 1:45)
+ {

+   dp[1,i]=d1[com[1,i],com[2,i]]+d1[com[3,i],com[4,i]]

+   dp[2,i]=d1[com[1,i],com[3,i]]+d1[com[2,i],com[4,i]]

+   dp[3,i]=d1[com[1,i],com[4,i]]+d1[com[3,i],com[2,i]]

+ }

> round(dp,5)


>
```



```
        [,1]    [,2]    [,3]    [,4]    [,5]    [,6]    [,7]    [,8]    [,9]   [,10]   [,11]   [,12]   [,13]   [,14]   [,15]
[1,] 0.00417 0.01181 0.01082 0.00612 0.00067 0.00939 0.01762 0.00794 0.00695 0.00224 0.00454 0.00552 0.01374 0.00128 0.00599
[2,] 0.00720 0.01181 0.01082 0.00612 0.01145 0.00939 0.01762 0.00794 0.00695 0.00224 0.00757 0.00552 0.01374 0.01098 0.00627
[3,] 0.00720 0.01123 0.01024 0.00553 0.01145 0.00880 0.01703 0.00735 0.00636 0.00165 0.00757 0.00493 0.01316 0.01098 0.00627
       [,16]   [,17]   [,18]   [,19]   [,20]   [,21]   [,22]   [,23]   [,24]   [,25]   [,26]   [,27]   [,28]   [,29]   [,30]
[1,] 0.01219 0.00272 0.00610 0.00500 0.01120 0.00173 0.00709 0.00649 0.00357 0.01180 0.00977 0.01800 0.00852 0.01333 0.01234
[2,] 0.01160 0.00955 0.01778 0.00528 0.01061 0.00856 0.01679 0.00591 0.00385 0.01208 0.00977 0.01800 0.01535 0.01333 0.01234
[3,] 0.01219 0.00955 0.01778 0.00528 0.01120 0.00856 0.01679 0.00649 0.00385 0.01208 0.00918 0.01741 0.01535 0.00971 0.00872
       [,31]   [,32]   [,33]   [,34]   [,35]   [,36]   [,37]   [,38]   [,39]   [,40]   [,41]   [,42]   [,43]   [,44]   [,45]
[1,] 0.00763 0.00993 0.01091 0.01913 0.00667 0.01138 0.01758 0.00810 0.01149 0.01039 0.01659 0.00711 0.01248 0.01188 0.00896
[2,] 0.00763 0.00218 0.01091 0.01913 0.01637 0.01166 0.00621 0.01494 0.02316 0.01067 0.00522 0.01395 0.02217 0.00052 0.00924
[3,] 0.00402 0.00993 0.00729 0.01552 0.01637 0.01166 0.01758 0.01494 0.02316 0.01067 0.01659 0.01395 0.02217 0.01188 0.00924
```

Interpretation:

Here, every triplet shows that, two of the distances are equal and larger than the third.

Hence, four-point condition is satisfied. Therefore, additivity property holds.

Therefore, the distance matrix found using distance function as difference between entropy of two sequences is additive.

Interpretations of a phylogenic tree:

i) Initially, sequence 1 and sequence 7 have evolved from a common ancestor (node19) with an evolutionary time equal to 0.00007 units. This means that sequences 1 and 7 have the highest degree of kinship among all the other sequences.

ii) Sequence 2 and node 19 (i.e. sequence 1 and sequence 7) have evolved from acommon ancestor (node 18) with an evolutionary time equal to 0.00018 units.

iii) Sequences 8 and sequence 3 have evolved from theancestor (node 13) with an evolutionary time of 0.00050 units.

iv)

v) Sequences 9 and node 17 have evolved from node 16 with evolutionary time equal to 0.00096 units.

vi) Sequences 4 and node 18 (i.e. sequence 4 and sequence 6) have evolved from node 15 with evolutionary time equal to 0.00088 units.

vii) Sequences 2 and node19 have evolved from node 18 with evolutionary time equal to 0.00018 units.

viii) node 15 and node 16 have evolved from node 14 with evolutionary time equal to 0.00260 units.

ix) node 14 and node 13 have evolved from node 12 with evolutionary time equal to 0.003800 units.

x) node 12 and sequence 10 have evolved from node 11 with evolutionary time equal to 0.00580 units.

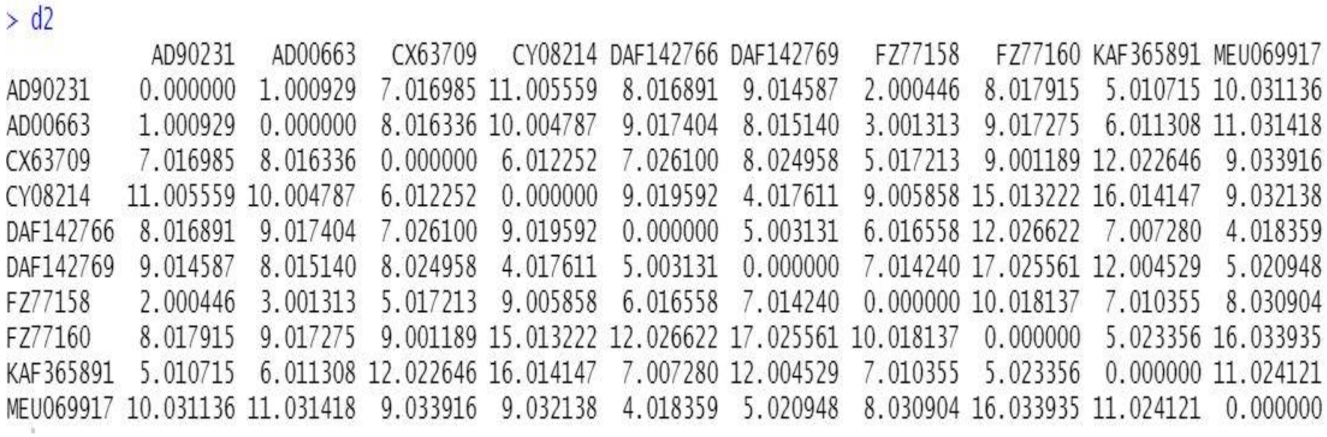
xi) Node 11 is the origin of evolution (first ancestor) for all the remaining sequences and Sequence 10 is most distantly related to all the other sequences, as it took maximumtime to evolve from node 11.

Euclidean Distance function:

If $X = (x_1, x_2, \dots \dots, x_n)$ and $Y = (y_1, y_2, \dots \dots, y_n)$ are two points in Euclidean n-space,then the distance (d) from X to Y or from Y to X is given by the following formulae;

$$distance = \sqrt{\sum_{i=0}^{n} (x_i - y_i)^2}$$

```
> dA=dist(A,method="euclidean",diag=TRUE,upper=TRUE)
> dG=dist(G,method="euclidean",diag=TRUE,upper=TRUE)
> dC=dist(C,method="euclidean",diag=TRUE,upper=TRUE)
> dT=dist(D,method="euclidean",diag=TRUE,upper=TRUE)
>
> d2=as.matrix(dA+dG+dC+dT)
> d2
```

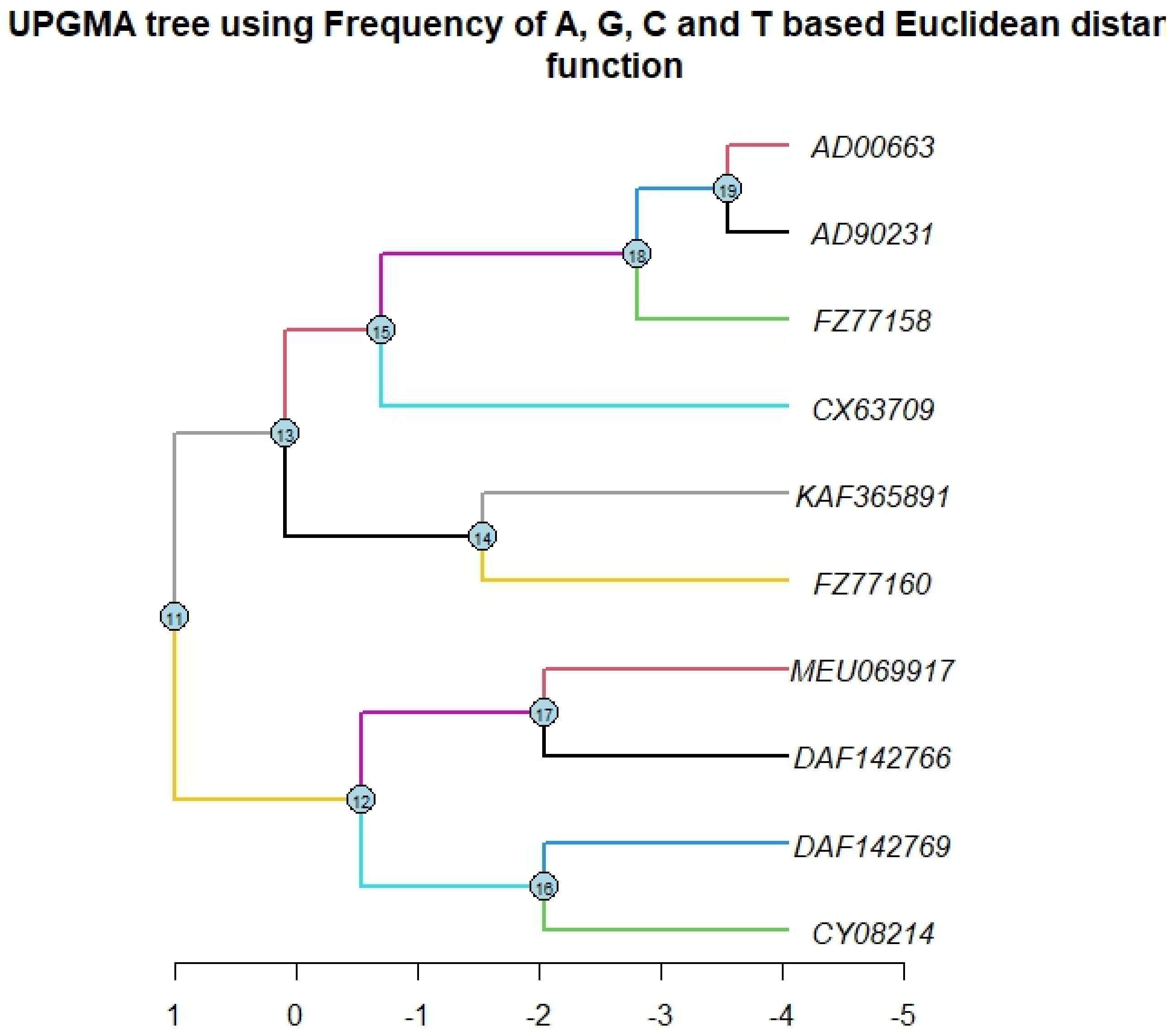


```
> d2
            AD90231   AD00663   CX63709   CY08214 DAF142766 DAF142769   FZ77158   FZ77160 KAF365891 MEU069917
AD90231    0.000000  1.000929  7.016985 11.005559  8.016891  9.014587  2.000446  8.017915  5.010715 10.031136
AD00663    1.000929  0.000000  8.016336 10.004787  9.017404  8.015140  3.001313  9.017275  6.011308 11.031418
CX63709    7.016985  8.016336  0.000000  6.012252  7.026100  8.024958  5.017213  9.001189 12.022646  9.033916
CY08214   11.005559 10.004787  6.012252  0.000000  9.019592  4.017611  9.005858 15.013222 16.014147  9.032138
DAF142766  8.016891  9.017404  7.026100  9.019592  0.000000  5.003131  6.016558 12.026622  7.007280  4.018359
DAF142769  9.014587  8.015140  8.024958  4.017611  5.003131  0.000000  7.014240 17.025561 12.004529  5.020948
FZ77158    2.000446  3.001313  5.017213  9.005858  6.016558  7.014240  0.000000 10.018137  7.010355  8.030904
FZ77160    8.017915  9.017275  9.001189 15.013222 12.026622 17.025561 10.018137  0.000000  5.023356 16.033935
KAF365891  5.010715  6.011308 12.022646 16.014147  7.007280 12.004529  7.010355  5.023356  0.000000 11.024121
MEU069917 10.031136 11.031418  9.033916  9.032138  4.018359  5.020948  8.030904 16.033935 11.024121  0.000000
```

This is the distance matrix obtained using Euclidean distance function based on frequency of A, G, C,T.

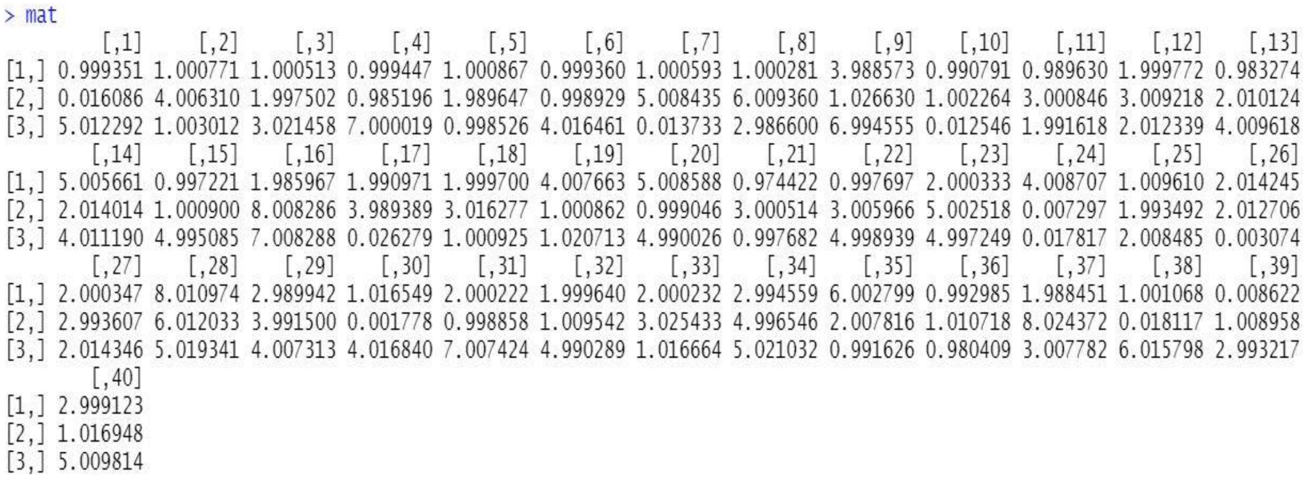
```
> tree=upgma(d2)
>
row.names(d1)=c("AD90231","AD00663","CX63709","CY08214","DAF142766","DAF142769","FZ77158","FZ77160","KAF365891","MEU069917")
> branch_name1=round(branching.times(tree),5)
> branch_name1
     11      12      13      14      15      16      17      18      19
5.05107 3.50948 4.13185 2.51168 3.34176 2.00881 2.00918 1.25044 0.50046
```

```
plot(tree,main="UPGMA tree using Frequency of A, G, C and T based Euclidean distance
function",type = "phylogram",adj = 0.5,edge.width = 2,edge.color = 1:10)
nodelabels(tree$node.label,cex=0.6,frame = "circle")
axisPhylo(side = 1,root.time = TRUE)
```



# ultrametric condition:

```
> D = as.matrix(d2)

> w = combn(1:10,3)

> w1 = matrix(0,3,120)

> nw = matrix(0,2,120)

> mini = c()
> ultcon2 = c()

> for(i in 1:120)

+ {

+   w1[1,i] = D[w[1,i],w[2,i]]

+   w1[2,i] = D[w[1,i],w[3,i]]

+   w1[3,i] = D[w[2,i],w[3,i]]

+  mini[i] = which.min(w1[,i])

+  nw[,i] = w1[-mini[i],i]


+  ultcon2[i] = abs(nw[1,i]-nw[2,i])

+  r=round(ultcon2,6)

+ }

> mat = matrix(r,nrow=3,byrow=TRUE)

> mat
```



```
> mat
         [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]     [,9]    [,10]    [,11]    [,12]    [,13]
[1,] 0.999351 1.000771 1.000513 0.999447 1.000867 0.999360 1.000593 1.000281 3.988573 0.990791 0.989630 1.999772 0.983274
[2,] 0.016086 4.006310 1.997502 0.985196 1.989647 0.998929 5.008435 6.009360 1.026630 1.002264 3.000846 3.009218 2.010124
[3,] 5.012292 1.003012 3.021458 7.000019 0.998526 4.016461 0.013733 2.986600 6.994555 0.012546 1.991618 2.012339 4.009618
        [,14]    [,15]    [,16]    [,17]    [,18]    [,19]    [,20]    [,21]    [,22]    [,23]    [,24]    [,25]    [,26]
[1,] 5.005661 0.997221 1.985967 1.990971 1.999700 4.007663 5.008588 0.974422 0.997697 2.000333 4.008707 1.009610 2.014245
[2,] 2.014014 1.000900 8.008286 3.989389 3.016277 1.000862 0.999046 3.000514 3.005966 5.002518 0.007297 1.993492 2.012706
[3,] 4.011190 4.995085 7.008288 0.026279 1.000925 1.020713 4.990026 0.997682 4.998939 4.997249 0.017817 2.008485 0.003074
        [,27]    [,28]    [,29]    [,30]    [,31]    [,32]    [,33]    [,34]    [,35]    [,36]    [,37]    [,38]    [,39]
[1,] 2.000347 8.010974 2.989942 1.016549 2.000222 1.999640 2.000232 2.994559 6.002799 0.992985 1.988451 1.001068 0.008622
[2,] 2.993607 6.012033 3.991500 0.001778 0.998858 1.009542 3.025433 4.996546 2.007816 1.010718 8.024372 0.018117 1.008958
[3,] 2.014346 5.019341 4.007313 4.016840 7.007424 4.990289 1.016664 5.021032 0.991626 0.980409 3.007782 6.015798 2.993217
        [,40]
[1,] 2.999123
[2,] 1.016948
[3,] 5.009814
```

This is the output of the ultrametric condition.

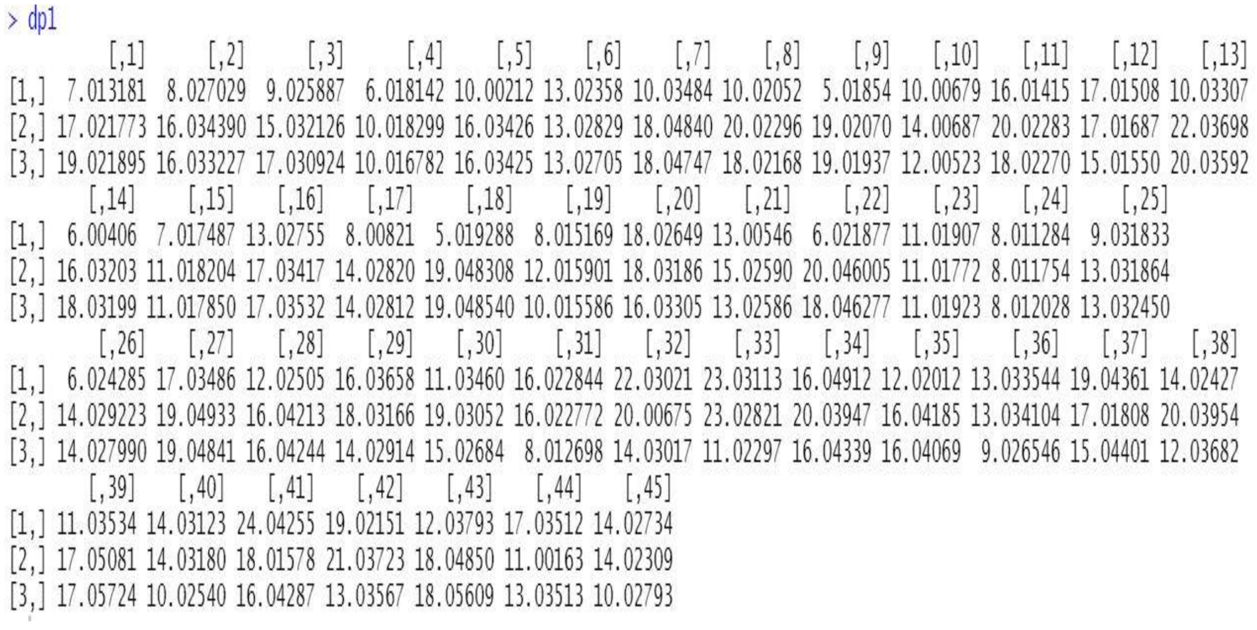
Interpretation:

Somewhere two of the distances among a triplet are equal and third is smaller than that but it is not seen in every triplet. Hence, ultrametric condition is not satisfied.

Hence, the phylogenetic tree topology using distance function as Euclidean distance based on frequency of A,G,C,T cannot said to be reliable.

## four-point condition:

```
> com=combn(1:10,4)
> dp1=matrix(0,3,45)
> for(i in 1:45)
+ {
+   dp1[1,i]=d2[com[1,i],com[2,i]]+d2[com[3,i],com[4,i]]
+   dp1[2,i]=d2[com[1,i],com[3,i]]+d2[com[2,i],com[4,i]]
+   dp1[3,i]=d2[com[1,i],com[4,i]]+d2[com[3,i],com[2,i]]+ }; >dp1
```



```
> dp1
          [,1]      [,2]      [,3]      [,4]     [,5]     [,6]     [,7]     [,8]     [,9]    [,10]    [,11]    [,12]    [,13]
[1,]  7.013181  8.027029  9.025887  6.018142 10.00212 13.02358 10.03484 10.02052  5.01854 10.00679 16.01415 17.01508 10.03307
[2,] 17.021773 16.034390 15.032126 10.018299 16.03426 13.02829 18.04840 20.02296 19.02070 14.00687 20.02283 17.01687 22.03698
[3,] 19.021895 16.033227 17.030924 10.016782 16.03425 13.02705 18.04747 18.02168 19.01937 12.00523 18.02270 15.01550 20.03592
        [,14]     [,15]    [,16]    [,17]     [,18]     [,19]    [,20]    [,21]     [,22]    [,23]    [,24]     [,25]
[1,]  6.00406  7.017487 13.02755  8.00821  5.019288  8.015169 18.02649 13.00546  6.021877 11.01907 8.011284  9.031833
[2,] 16.03203 11.018204 17.03417 14.02820 19.048308 12.015901 18.03186 15.02590 20.046005 11.01772 8.011754 13.031864
[3,] 18.03199 11.017850 17.03532 14.02812 19.048540 10.015586 16.03305 13.02586 18.046277 11.01923 8.012028 13.032450
         [,26]    [,27]    [,28]    [,29]    [,30]     [,31]    [,32]    [,33]    [,34]    [,35]     [,36]    [,37]    [,38]
[1,]  6.024285 17.03486 12.02505 16.03658 11.03460 16.022844 22.03021 23.03113 16.04912 12.02012 13.033544 19.04361 14.02427
[2,] 14.029223 19.04933 16.04213 18.03166 19.03052 16.022772 20.00675 23.02821 20.03947 16.04185 13.034104 17.01808 20.03954
[3,] 14.027990 19.04841 16.04244 14.02914 15.02684  8.012698 14.03017 11.02297 16.04339 16.04069  9.026546 15.04401 12.03682
        [,39]    [,40]    [,41]    [,42]    [,43]    [,44]    [,45]
[1,] 11.03534 14.03123 24.04255 19.02151 12.03793 17.03512 14.02734
[2,] 17.05081 14.03180 18.01578 21.03723 18.04850 11.00163 14.02309
[3,] 17.05724 10.02540 16.04287 13.03567 18.05609 13.03513 10.02793
```

This is the output of four-point condition.

Interpretation:

Here, most of the triplets shows that, two of the distances are equal and larger than the third.

Hence, four-point condition is satisfied. Therefore, additivity property holds.

Therefore, the distance matrix found using distance function as Euclidean distance based on frequency of A, G, C, T is additive.

Interpretations of a phylogenic tree:

i) Initially, sequence AD00663 and sequence AD 90231 have evolved from a common ancestor (node19)with the evolutionary time equal to 0.50046 units.

ii) Sequence FZ77158 and Node 19 have evolved from a common ancestor (Node 18) with theevolutionary time equal to 1.25044 units.

iii) Sequence DAF142769 and sequence CY08214 have evolved from common ancestor (Node 16) with theevolutionary time equal to 2.00881 units.

iv) Sequence MEU069917 and sequence DAF14276 6have evolved from common ancestor (Node 17) withevolutionary time equal to 2.00918 units.

v) Sequence KAF365891 and Sequence FZ77160 have evolved from common ancestor (Node 14) withevolutionary time 2.51168 units.

vi) Node 14 and Node 15 have evolved from common ancestor (Node 13) with evolutionary time equal to 4.13185 units.

vii) Node 17 and Node 16 have evolved from common ancestor (Node 12) with evolutionary time equal to 3.50948 unit.

viii) Node 12 and Node 13 have evolved from common ancestor (Node 11) with evolutionary time equal to 5.05107 units.

Distance function of our choice:

We have chosen Supremum Norm distance function to plot phylogenic tree.

Sup Norm distance function:

Suppose $X = (x_1, x_2, \dots \dots, x_n)$ and $Y = (y_1, y_2, \dots \dots, y_n)$ are two sequences, then the distance (d) from X to Y or from Y to X is given by the following formula;

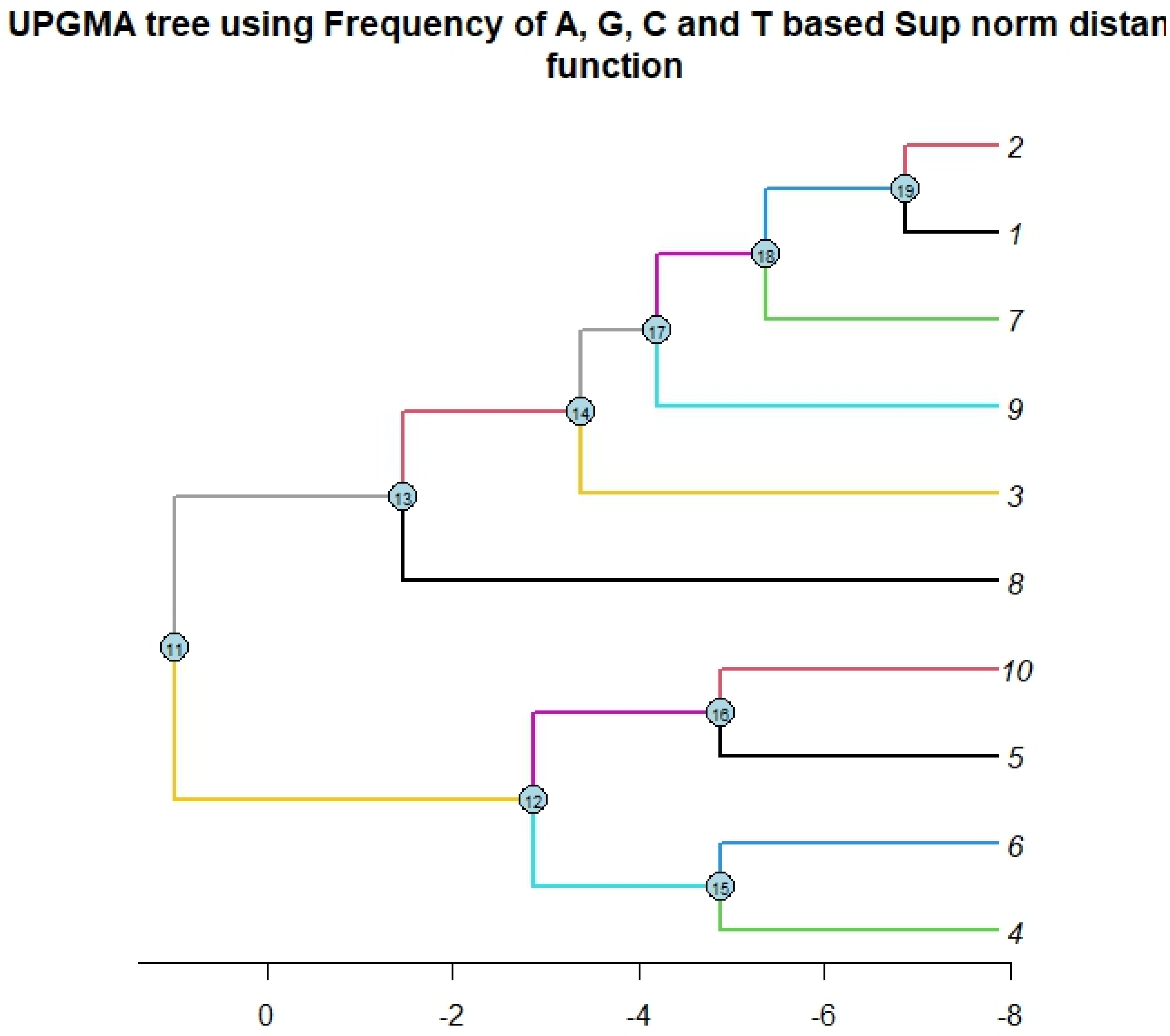
$$\underset{i}{(}X, Y) = \max|x_i - y_i|$$

```
> dA=dist(A,method="maximum",diag=TRUE,upper=TRUE)

> dG=dist(G,method="maximum",diag=TRUE,upper=TRUE)

> dC=dist(C,method="maximum",diag=TRUE,upper=TRUE)

> dT=dist(T,method="maximum",diag=TRUE,upper=TRUE)

> d3=as.matrix(dA+dG+dC+dT)

> d3

> d3
   1  2  3  4  5  6  7  8  9 10
1   0  2  8 16 16 16  4 12  6 20
2   2  0 10 16 18 16  6 12  8 22
3   8 10  0 10 14 14  6 14 12 18
4  16 16 10  0 12  6 12 24 20 14
5  16 18 14 12  0  6 12 24 14  6
6  16 16 14  6  6  0 12 28 18  8
7   4  6  6 12 12 12  0 16  8 16
8  12 12 14 24 24 28 16  0 10 30
9   6  8 12 20 14 18  8 10  0 20
10 20 22 18 14  6  8 16 30 20  0
```

>

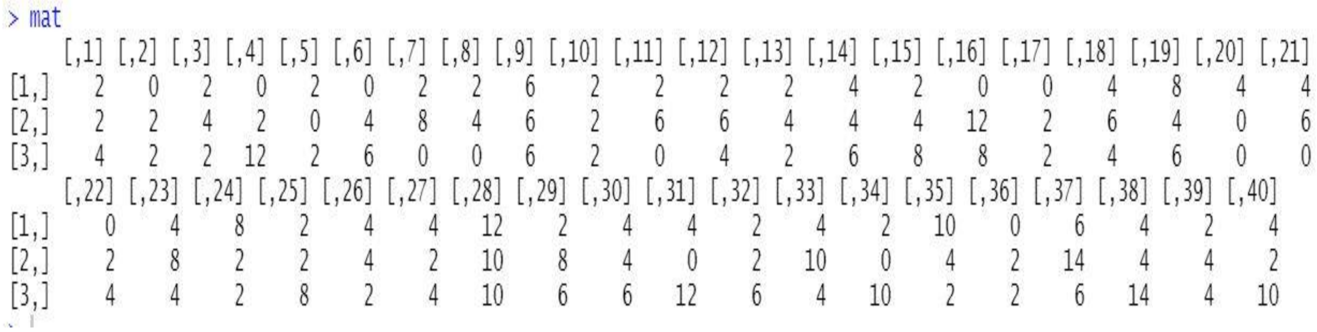
This is distance matrix based on Sup norm distance.

```
> tree=upgma(d3)

> row.names(d1)=c("I1","I2","I3","I4","I5","I6","I7","I8","I9","B1")

> branch_name1 = round(branching.times(tree),5)

> branch_name1
     11      12      13      14      15      16      17      18      19
8.87500 5.00000 6.40000 4.50000 3.00000 3.00000 3.66667 2.50000 1.00000

> plot(tree,main="UPGMA tree using Frequency of A, G, C and T based Sup norm distance
function",type = "phylogram",adj = 0.5,edge.width = 2,edge.color = 1:10)

> nodelabels(tree$node.label,cex=0.6,frame = "circle")

> axisPhylo(side = 1,root.time = TRUE)
```



UPGMA tree using Frequency of A, G, C and T based Sup norm distan function

```
## ultrametric condition

> D = as.matrix(d3)

> w = combn(1:10,3)

> w1 = matrix(0,3,120)

> nw = matrix(0,2,120)

> mini = c()
> ultcon3 = c()
> for(i in 1:120)
+ {
+   w1[1,i] = D[w[1,i],w[2,i]]
+   w1[2,i] = D[w[1,i],w[3,i]]
+   w1[3,i] = D[w[2,i],w[3,i]]
+  mini[i] = which.min(w1[,i])
+  nw[,i] = w1[-mini[i],i]
+  ultcon3[i] = abs(nw[1,i]-nw[2,i])
+  r=round(ultcon3,6)
+ }
> mat = matrix(r,nrow=3,byrow=TRUE)
> mat
```



```
> mat
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21]
[1,]    2    0    2    0    2    0    2    2    6     2     2     2     2     4     2     0     0     4     8     4     4
[2,]    2    2    4    2    0    4    8    4    6     2     6     6     4     4     4    12     2     6     4     0     6
[3,]    4    2    2   12    2    6    0    0    6     2     0     4     2     6     8     8     2     4     6     0     0
     [,22] [,23] [,24] [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37] [,38] [,39] [,40]
[1,]     0     4     8     2     4     4    12     2     4     4     2     4     2    10     0     6     4     2     4
[2,]     2     8     2     2     4     2    10     8     4     0     2    10     0     4     2    14     4     4     2
[3,]     4     4     2     8     2     4    10     6     6    12     6     4    10     2     2     6    14     4    10
```

This is the output of the ultrametric condition.

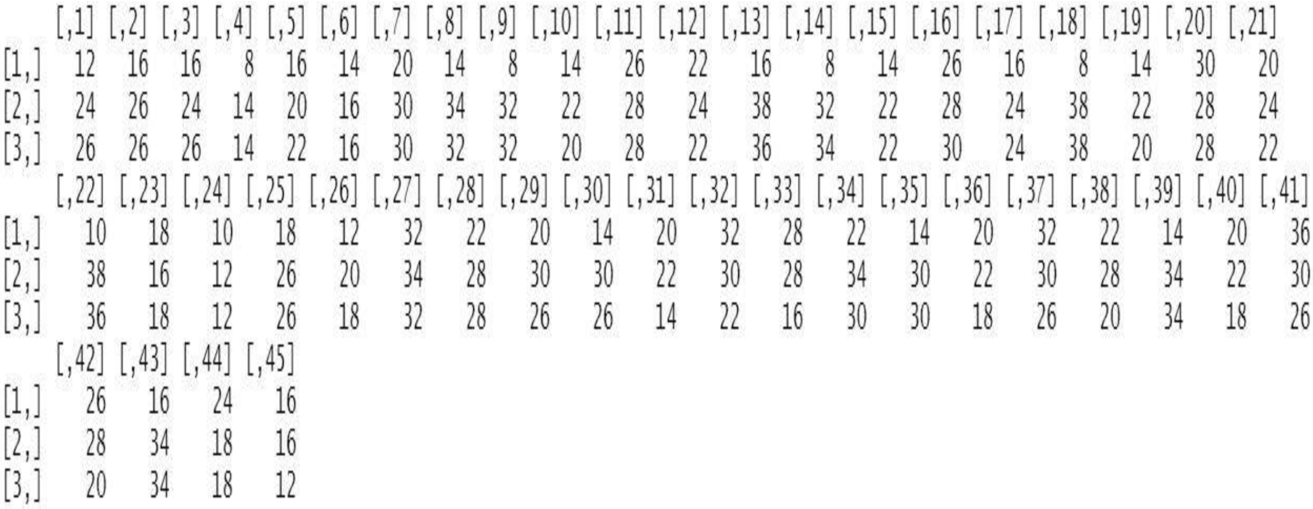
Interpretation:

Somewhere two of the distances among a triplet are equal and third is smaller than that but it is not seen in every triplet. Hence, ultrametric condition is not satisfied.

Hence, the phylogenetic tree topology using Sup norm distance function cannot said to be reliable.

## Four- point condition:

```
> com=combn(1:10,4)

> dp2=matrix(0,3,45)

> for(i in 1:45)

+ {

+  dp2[1,i]=d3[com[1,i],com[2,i]]+d3[com[3,i],com[4,i]]

+  dp2[2,i]=d3[com[1,i],com[3,i]]+d3[com[2,i],com[4,i]]

+  dp2[3,i]=d3[com[1,i],com[4,i]]+d3[com[3,i],com[2,i]]

+ }

> dp2
```



```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21]
[1,]   12   16   16    8   16   14   20   14    8    14    26    22    16     8    14    26    16     8    14    30    20
[2,]   24   26   24   14   20   16   30   34   32    22    28    24    38    32    22    28    24    38    22    28    24
[3,]   26   26   26   14   22   16   30   32   32    20    28    22    36    34    22    30    24    38    20    28    22
     [,22] [,23] [,24] [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37] [,38] [,39] [,40] [,41]
[1,]    10    18    10    18    12    32    22    20    14    20    32    28    22    14    20    32    22    14    20    36
[2,]    38    16    12    26    20    34    28    30    30    22    30    28    34    30    22    30    28    34    22    30
[3,]    36    18    12    26    18    32    28    26    26    14    22    16    30    30    18    26    20    34    18    26
     [,42] [,43] [,44] [,45]
[1,]    26    16    24    16
[2,]    28    34    18    16
[3,]    20    34    18    12
```

This is the output of Four-point condition.

Interpretation:

Here, most of the triplets shows that, two of the distances are equal and larger than the third.

Hence, four-point condition is satisfied. Therefore, additivity property holds.

Therefore, the distance matrix found using Sup norm distance function is additive.

Interpretations of a phylogenic tree:

i) Initially, sequence 1 and sequence 2 have evolved from a common ancestor (node19) with the evolutionary time equal to 1 units.

ii) Sequence 7 and Node 19 have evolved from a common ancestor (Node 18) with the evolutionary time equal to 2.50000 units.

iii) Sequence 6 and sequence 4 have evolved from common ancestor (Node 15) with evolutionary time equal to 3 units.

iv) Node 18 and Sequence 9 have evolved from common ancestor (Node 17) with evolutionary time 3.66667 units.

v) Sequence 17 and Sequence 3 have evolved from common ancestor (Node 14) withevolutionary time 4.50000 units.

vi) Node 16 and Node 15 have evolved from common ancestor (Node 12) with evolutionary time 5.0000 units

vii) Sequence 10 and sequence 5 have evolved from a common ancestor (Node 16) with theevolutionary time equal to 3 units.

viii) Node 14 and sequence 3 have evolved from common ancestor (Node 13) withevolutionary time 6.40000 units.

ix) Node 13 and node 12 have evolved from common ancestor (Node 12) with evolutionary time 8.87500 units.

x) **Node 11 is the origin of evolution (first ancestor) for all the remaining sequences and Sequence 10 is most distantly related to all the other sequences, as it took maximum timeto evolve from node 11.**

Conclusions:

We have observed that the tree topology by considering the Sup norm distance is same as that given by Euclidean distance. While this tree topology is different than from tree topology where distance function used is based on the difference between entropies.

Since, the four-point condition is satisfied by distance matrix formed using difference between entropies, Euclidean distance and sup norm distance functions. Therefore, those three distance matrices are additive. But the ultrametric condition isnot satisfied.

Hence, due to additivity to the distance matrices we can get the UPGMA phylogenetic trees but those trees may not be reliable because of un-satisfied ultrametriccondition.

Question 3)

Explain how do you use mutual information content to obtain tree topology. Using your suggested algorithm, obtain tree topology for your data.

Solution:

Use of Mutual Information Content in obtaining tree topology:

We review a conceptually very simple algorithm for hierarchical clustering called in the following the mutual information clustering (MIC) algorithm. It uses mutual information (MI) as a similarity measure and exploits its grouping property: The MI between three objects X,Y, and Z is equal to the sum of the MI between X and Y, plus the MI between Z and the combined object (XY).

More precisely, we propose the following scheme for clustering n objects with MIC:

(1) Compute a proximity matrix based on pairwise mutual informations; assign n clusters such that each cluster contains exactly one object;

(2) find the two closest clusters i and j;

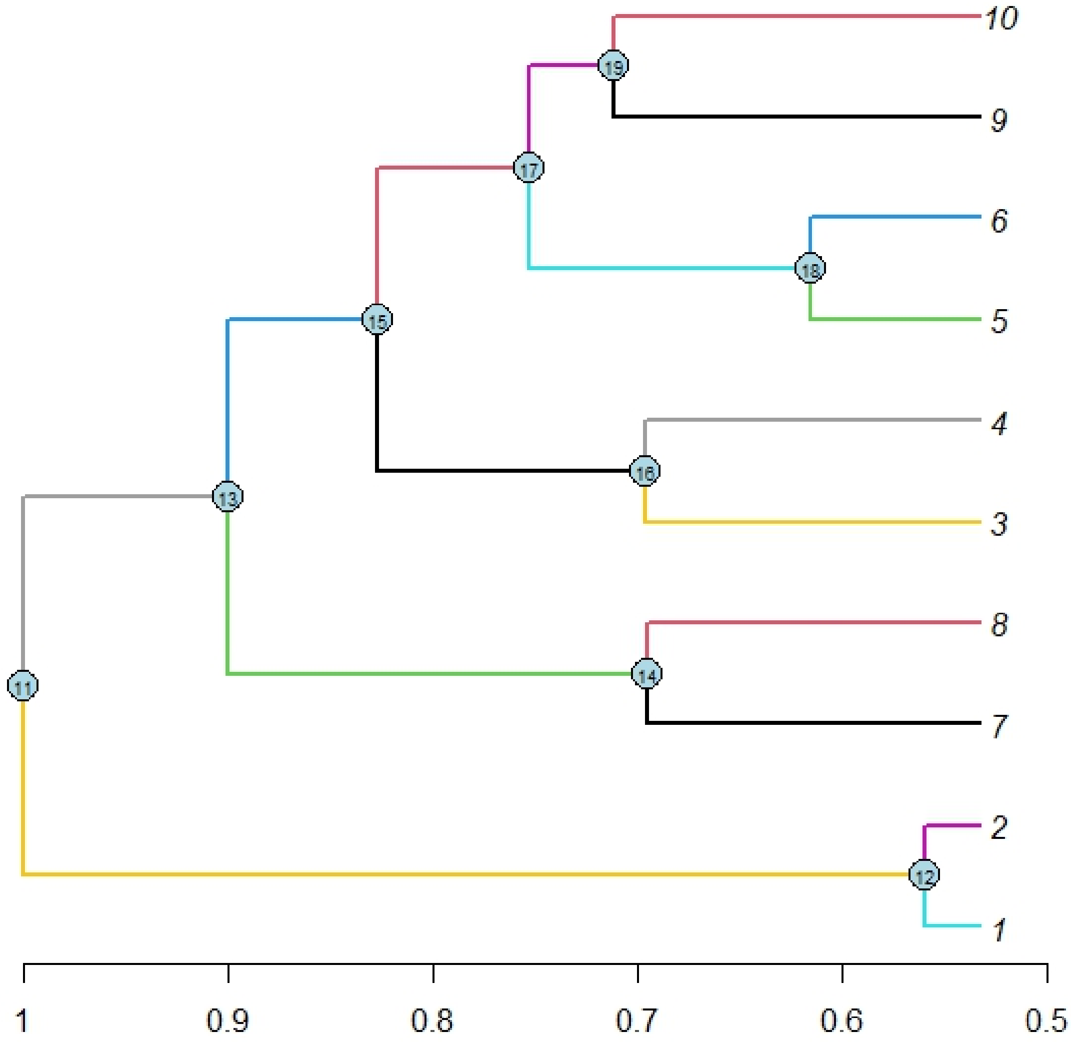
(3) create a new cluster (ij) by combining i and j;

(4) delete the lines/columns with indices i and j from the proximity matrix, and add one line/column containing the proximities between cluster (ij) and all other clusters;

(5) if the number of clusters is still > 2, goto (2); else join the two clusters and stop.

Obtaining Tree Topology using above algorithm:

```
> D_mat = dist(MIC4,diag = TRUE, upper = TRUE)

> Tree= upgma(D_mat)

> branch_name = round(branching.times(Tree),5)
branch_name
     11      12      13      14      15      16      17      18      19
0.46671 0.02615 0.36693 0.16247 0.29352 0.16260 0.21989 0.08247 0.17856

> plot(Tree,type = "phylogram",adj = 0.5,edge.width = 2,edge.color = 1:10)

>   nodelabels(Tree$node.label,cex=0.6,frame = "circle")

> axisPhylo(side = 1,root.time = TRUE)
```



Interpretations of a phylogenic tree:

i) Initially, sequence 10 and sequence 9 have evolved from a common ancestor (node19) with the evolutionary time equal to 0.17856 units. This means that sequences 4 and 6 have the highest degree of kinship among all the other sequences.

ii) Sequence 5 and sequence 6 have evolved from a common ancestor (node 18) with the evolutionary time equal to 0.08247 units.

iii) Node 18 and node 19 have evolved from the ancestor (node 17) with the evolutionary time 0.21989 units.

iv) Sequences 4 and sequence 3 have evolved from node 16 with evolutionary time equal to 0.16260 units.

v) node 16 (i.e. sequence 4 and sequence 6) and node 17 (i.e. sequence 5 and node 18) have evolved from node 14 with evolutionary time equal to 0.016259 units.

vi) Sequences 7 and sequence 8 have evolved from node 14 with evolutionary time equal to 0.16247 units.

vii) Sequence 1 and sequence 2 have evolved from node 12 with evolutionary time equal to 0.02615 units.

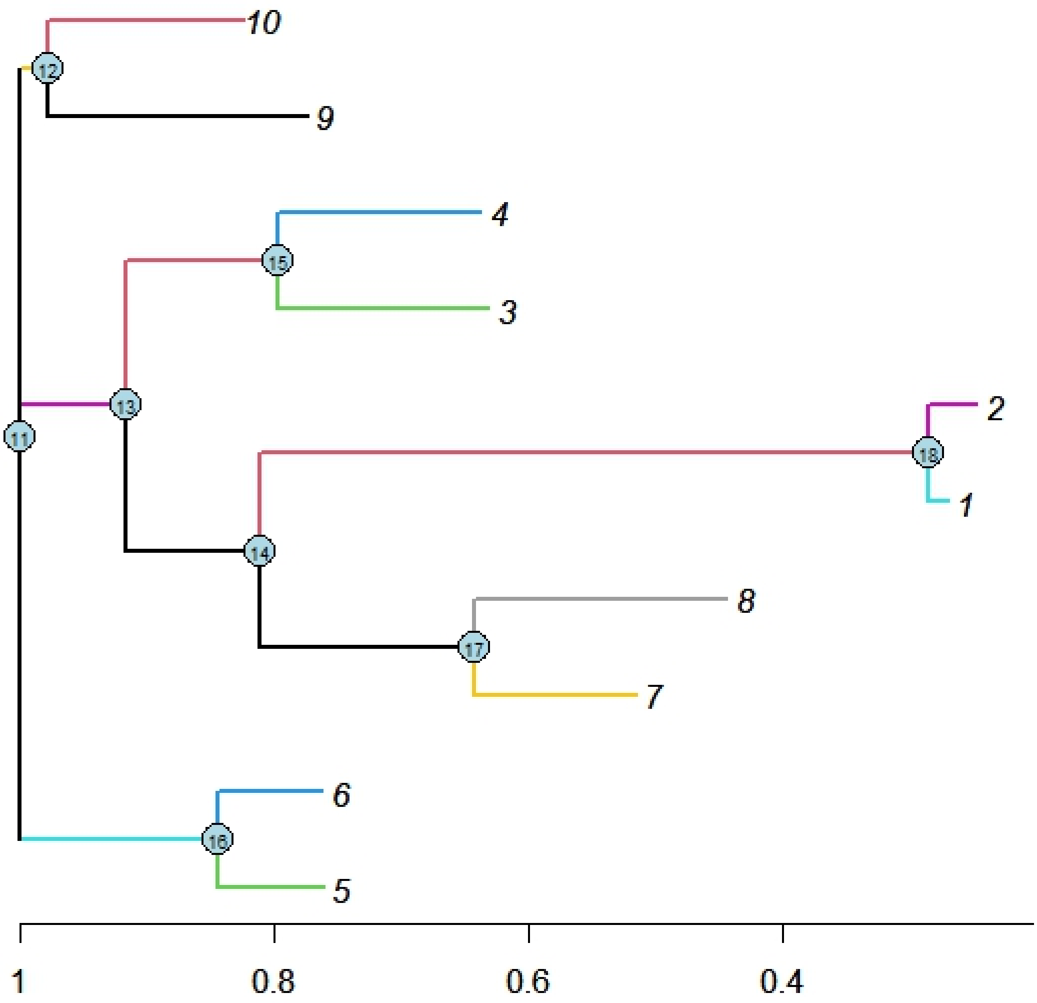
viii) Node 16 and node 17 have evolved from node 15 with evolutionary time equal to 0.29352 units.

ix) node 14 and node 15 have evolved from node 13 with evolutionary time equal to 0.36693 units.

x) node 13 and node 12 have evolved from node 13 with evolutionary time equal to 0.46671 units.

xi) Node 11 is the origin of evolution (first ancestor).

```
> nj_algorithm = NJ(D_mat)

> branch_name = round(branching.times(nj_algorithm),5)
>branch_name
11 12 13 14 15 16 17 18
 0  0  0  0  0  0  0 -1

> plot(nj_algorithm,type = "phylogram",adj = 0.5,edge.width = 2,edge.color = 1:10)

> nodelabels(nj_algorithm$node.label,cex=0.6,frame = "circle")

> axisPhylo(side = 1,root.time = TRUE)
```



Interpretations of a phylogenic tree:

i) Sequence 10 and sequence 10 have evolved from a common ancestor (node12)

ii) Sequence 4 and Sequence 3 have evolved from common ancestor (Node 15).

iii) Sequence 1 and sequence 2 have evolved from common ancestor (Node 18).

iv) Sequence 7 and Sequence 8 have evolved from common ancestor (Node 17).v)

v) Sequence 5 and sequence 6 have evolved from common ancestor (Node 16).

vi) Node 18 and node 17 have evolved from common ancestor (node 14).

vii) Node 15 and node 14 have evolved from common ancestor (Node 13).

viii) vii) Node 15 and node 13 and node 14 have evolved from common ancestor (Node 11).

ix) Node 11 seems to be the starting point of the evolutionary process.

Question 4:

Select three distance functions of your choice. Obtain the distance matrix for the each one of them. Verify which distance function satisfies ultrametric condition. Using N-J method obtain tree topology corresponding to each distance function. Comment on the result.

Solution:

Distance function:

In mathematics, a metric or distance function is a function that gives a distance between each pair of point elements of a set. A set with a metric is called a metric space. A metric induces a topology on a set, but not all topologies can be generated by a metric.
A topological space whose topology can be described by a metric is called metrizable.

A metric on a set *X* is a function (called *distance function* or simply *distance*) ,

$d: X \times X \rightarrow [0, \infty)$,

And following 3 axioms are satisfied:

1. $(x, y) = 0 \Leftrightarrow x = y$

2. $(x, y) = (y, x)$

1. $(x, y) \leq (x, z) + (z, y)$

Ultrametric condition:

The distances d_{ij} are said to be ultrametric if for every triplet of sequences x^i, x^j, x^k the distances d_{ij}, d_{ik}, d_{jk} either all are equal or two are equal and remaining one is smaller. Satisfaction of ultrametric condition implies constant evolution rate.

Neighbor – Joining Method:

The initial tree topology is a star.It reconstructs the unrooted phylogenetic tree with branch lengths using minimum evolution criterion that minimizes the lengths of tree.

It does not assume the constancy of substitution rates across sites and does not require the data to be ultrametric, unlike UPGMA. Hence, this method is more appropriate for the sites with variable rates of evolution. Input distance matrix is modified such that the distance between every pair of OTUs is adjusted using their average divergence from remaining OTUs.

Algorithm:

Step 1. Input the sequences. Step 2. Initialize all d_{ij} = 0 for all i = 1,2,3, … N & j = 1,2,3, … N (N = #of sequences)

Step 3. Choose an optimal distance function.

Step 4. Compute distance matrix D = ((d_{ij})) Step 5. Calculate corrected D^* = ((D_ij))

D_{ij} = d_{ij}– (r_i + r_j) where r_i = (1/N-2)*∑ The OPU's i and j will be grouped together if Dij = Min Dkl , call this as (N+1) (k,l)

Step 6. Compute branch lengths,

$$l_{N+1,} = (d_{ij} - r_j + r_i)/2$$

$$l_{N+1,} = (d_{ij} - r_i + r_j)/2$$

Step 7. Goto step 4 with applying rule of computing distance between composite OTU's,

$$_{(i,j)(k,l)} = (d_{ik} + d_{il} + d_{jk} + d)/4$$

Suppose xm,xn are two OTU's with sizes t_1 and t_2 then,

$$d_{mn} = \sum d_{ij}/(t_1 + t_2)$$

for $t_1 = t_2$

or

$$d_{mn} = \sum d_{ij}/(t_1 + t_2 - k)$$

for $k = t_2 - t_1$

Step 8. Represent diagrammatically.

Step 9. Stop.

R code:

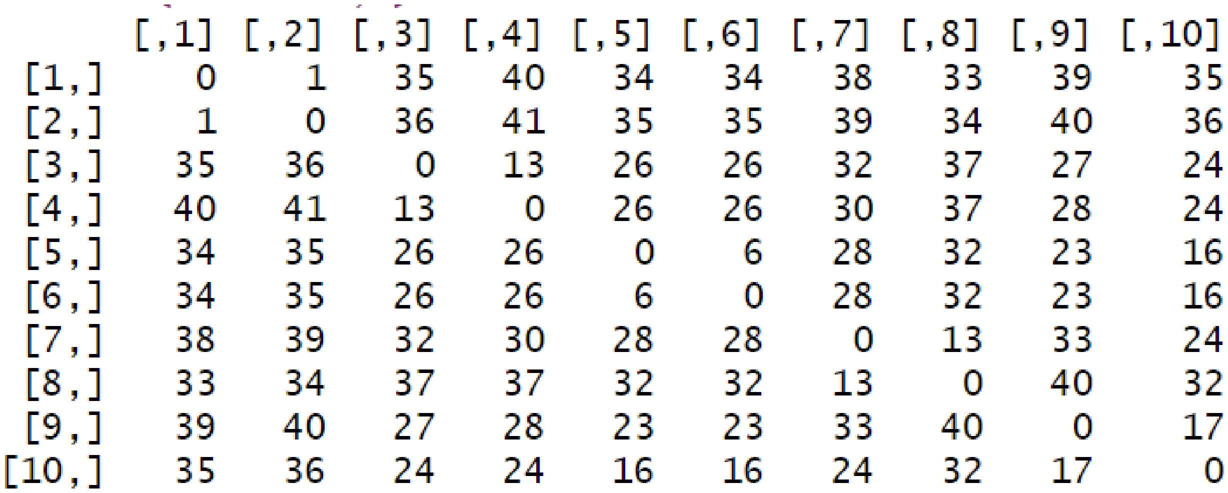
Q4

Distance Function 1: Hamming Diatance

The above code is for importing the sequences in R studios.

We created a distance function based on Hamming Distance Function. It is a mono-levelfunction with 4 terms under a square root

```
> Dmatrix1=matrix(0,nrow=10,ncol = 10)
> for(i in 1:10)
+ {
+   for(j in 1:10)
+   {
+     Dmatrix1[i,j]=sum(unlist(S[[i]])!=unlist(S[[j]]))
+   }
+ }
> round(Dmatrix1,2)
```



```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    0    1   35   40   34   34   38   33   39    35
 [2,]    1    0   36   41   35   35   39   34   40    36
 [3,]   35   36    0   13   26   26   32   37   27    24
 [4,]   40   41   13    0   26   26   30   37   28    24
 [5,]   34   35   26   26    0    6   28   32   23    16
 [6,]   34   35   26   26    6    0   28   32   23    16
 [7,]   38   39   32   30   28   28    0   13   33    24
 [8,]   33   34   37   37   32   32   13    0   40    32
 [9,]   39   40   27   28   23   23   33   40    0    17
[10,]   35   36   24   24   16   16   24   32   17     0
```

We created a distance function based on Hamming Distance Function.

The above matrix is the distance matrix that we obtained using the distance function defined.

Now for verifying the ultrametric condition, we chose 3 triplets (x^1, x^5, x^8), (x^2, x^6, x^9), (x^3, x^7, x^{10}).

We used this method because we observed that the ultrametric condition does not hold for them.

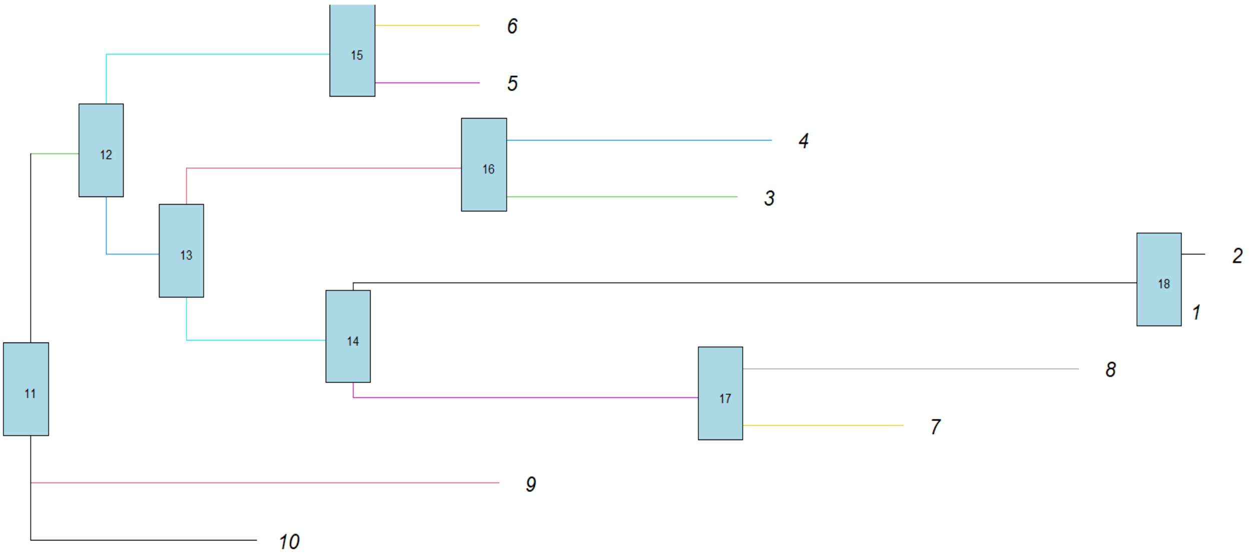
```
> ## First triplet (x1,x5,x8)
> c(Dmatrix1[1,5],Dmatrix1[1,8],Dmatrix1[5,8])
[1] 34 33 32

> ## Second triplet (x2,x6,x9)
> c(Dmatrix1[2,6],Dmatrix1[2,9],Dmatrix1[6,9])
[1] 35 40 23

> ## Third triplet (x3,x7,x10)
> c(Dmatrix1[3,7],Dmatrix1[3,10],Dmatrix1[7,10])
[1] 32 24 24
```

Interpretation:

As we can see, none of the distances are equal. This implies that the distance function does not satisfy the ultrametric condition. Therefore, the tree obtained from above distance function may not be as reliable.

NJ Method:

```
  plot(nj(Dmatrix1),type = "phylogram",adj = 0.5,edge.width =0.6,edge.color =
1:10,main="Topological Tree Using NJ Method(Distance Function 1)")


  nodelabels(nj(Dmatrix1)$node.label,cex=0.6,frame = "rect")
```



Topological Tree Using NJ Method (Distance Function 1)

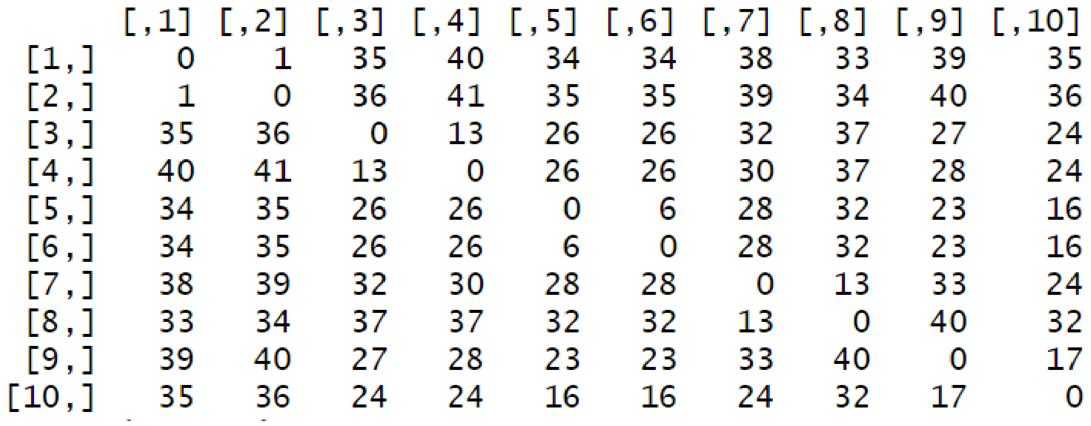
Interpretation:

Sequence 6 and Sequence 5 have evolved from common ancestor (Node 15).
Sequence 3 and Sequence 4 have evolved from common ancestor (Node 16).
Sequence 2 and Sequence 1 have evolved from common ancestor (Node 18).
Sequence 8 and Node 7 have evolved from common ancestor (Node 17).
Node 15 and Node 13 have evolved from common ancestor (Node12).
Sequence 16 and Node 13 have evolved from common ancestor
(Node14).
sequence 9 and sequence 10 and node 12 have evolved from common
ancestor (Node11).
All the evolutions started at Node 11.

Distance function 2: Raw Distance Function

The next distance function is from *dist.dna()* from package *ape* of R program. It is the *raw* distance function. This is simply the proportion or the number of sites that differ between each pair of sequences. This may be useful to draw “saturation plots”.

```
> # Define function for raw distance
> raw.dist <- function(seq1, seq2) {
+   d <- sum(seq1 != seq2)
+   return(d)
+ }
> # Initialize Dmatrix2
> Dmatrix2 <- matrix(0, nrow = 10, ncol = 10)
> # Calculate raw distance between sequences
> for (i in 1:10) {
+   for (j in 1:10) {
+     if (i == j) {
+       Dmatrix2[i, j] <- 0
+     } else {
+       Dmatrix2[i, j] <- raw.dist(S[[i]], S[[j]])
+     }
+   }
+ }
> Dmatrix2
```



```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    0    1   35   40   34   34   38   33   39    35
 [2,]    1    0   36   41   35   35   39   34   40    36
 [3,]   35   36    0   13   26   26   32   37   27    24
 [4,]   40   41   13    0   26   26   30   37   28    24
 [5,]   34   35   26   26    0    6   28   32   23    16
 [6,]   34   35   26   26    6    0   28   32   23    16
 [7,]   38   39   32   30   28   28    0   13   33    24
 [8,]   33   34   37   37   32   32   13    0   40    32
 [9,]   39   40   27   28   23   23   33   40    0    17
[10,]   35   36   24   24   16   16   24   32   17     0
```

The above is the distance matrix obtained.

```
> ## First triplet (x1,x5,x8)
> c(Dmatrix2[1,5],Dmatrix2[1,8],Dmatrix2[5,8])
[1] 34 33 32


> ## Second triplet (x2,x6,x9)
> c(Dmatrix2[2,6],Dmatrix2[2,9],Dmatrix2[6,9])
[1] 35 40 23

> ## Third triplet (x3,x7,x10)
> c(Dmatrix2[3,7],Dmatrix2[3,10],Dmatrix2[7,10])
[1] 32 24 24
```

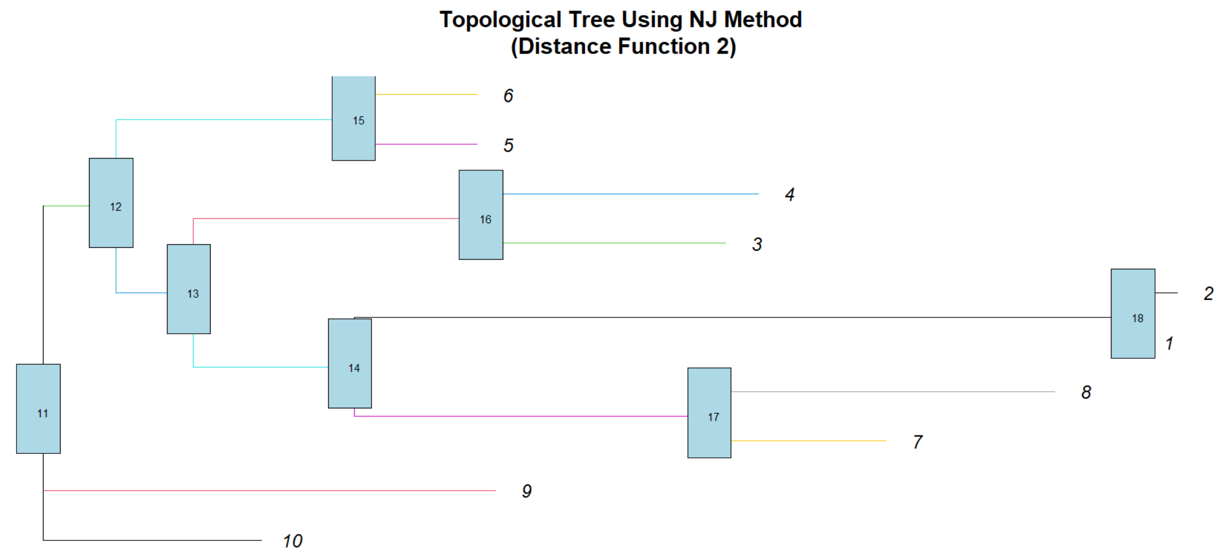
Interpretation:

As we can see, none of the distances are equal. This implies that the distance function does not satisfy the ultrametric condition. Therefore the tree obtained from above distance function may not be as reliable.

NJ Method:

```
plot(nj(Dmatrix2),type = "phylogram",adj = 0.5,edge.width =0.6,edge.color =
1:10,main="Topological Tree Using NJ Method(Distance Function 2)")
```

```
nodelabels(nj(Dmatrix2)$node.label,cex=0.6,frame = "rect")
```



Interpretation:

Sequence 6 and Sequence 5 have evolved from common ancestor (Node 15).

Sequence 3 and Sequence 4 have evolved from common ancestor (Node 16).

Sequence 2 and Sequence 1 have evolved from common ancestor (Node 18).

Sequence 8 and Node 7 have evolved from common ancestor (Node 17).

Node 15 and Node 13 have evolved from common ancestor (Node12).

Sequence 16 and Node 13 have evolved from common ancestor (Node14).

sequence 9 and sequence 10 and node 12 have evolved from common ancestor (Node11).

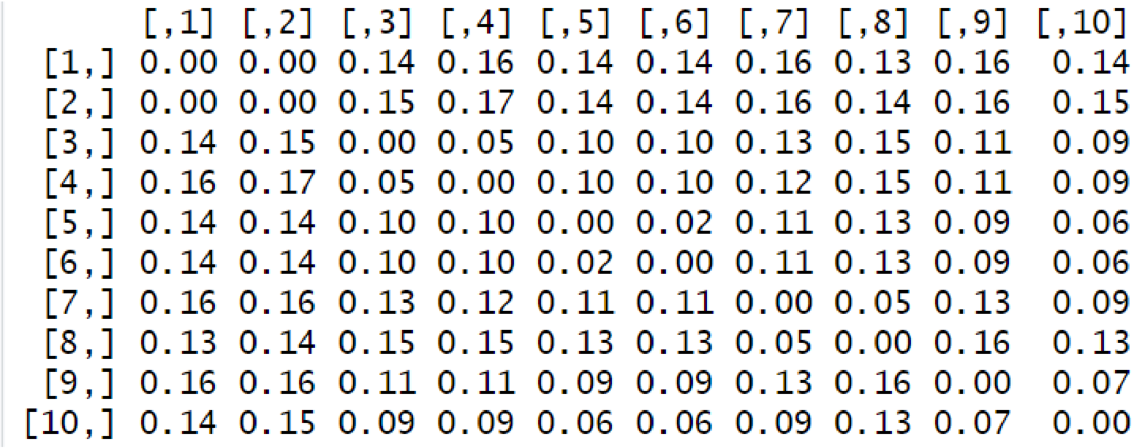
All the evolutions started at Node 11.

Distance function 3: JC69 Distance Function

The next distance function is from *dist.dna()* from package *ape* of R program. It is the *JC69* distance function. This model was developed by Jukes and Cantor (1969). It assumes that all substitutions (i.e. a change of a base by another one) have the same probability. This probability is the same for all sites along the DNA sequence. This last assumption can be relaxed by assuming that the substition rate varies among site following a gamma distribution which parameter must be given by the user. By default, no gamma correction is applied. Another assumption is that the base frequencies are balanced and thus equal to 0.25.

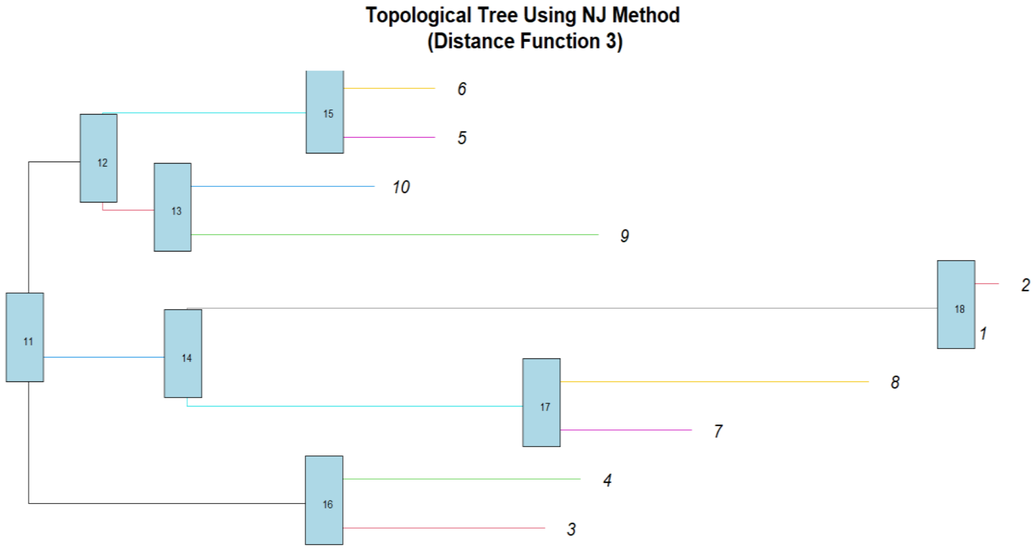
```
> ########## Define function for JC69 distance
> # Load required packages
> library(seqinr)


> library(ape)
> # Define function for JC69 distance
> JC69.dist <- function(seq1, seq2) {
+   p <- sum(seq1 != seq2) / length(seq1)
+   d <- -0.75 * log(1 - 4/3 * p)
+   return(d)
+ }
> # Initialize Dmatrix3
> Dmatrix3 <- matrix(0, nrow = 10, ncol = 10)
> # Calculate JC69 distance between sequences
> for (i in 1:10) {
+   for (j in 1:10) {
+     if (i == j) {
```

```
+       Dmatrix3[i, j] <- 0
+     } else {
+       Dmatrix3[i, j] <- JC69.dist(S[[i]], S[[j]])
+     }
+   }
+ }
> # View the Dmatrix1
> round(Dmatrix3,2)
```



```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,] 0.00 0.00 0.14 0.16 0.14 0.14 0.16 0.13 0.16  0.14
 [2,] 0.00 0.00 0.15 0.17 0.14 0.14 0.16 0.14 0.16  0.15
 [3,] 0.14 0.15 0.00 0.05 0.10 0.10 0.13 0.15 0.11  0.09
 [4,] 0.16 0.17 0.05 0.00 0.10 0.10 0.12 0.15 0.11  0.09
 [5,] 0.14 0.14 0.10 0.10 0.00 0.02 0.11 0.13 0.09  0.06
 [6,] 0.14 0.14 0.10 0.10 0.02 0.00 0.11 0.13 0.09  0.06
 [7,] 0.16 0.16 0.13 0.12 0.11 0.11 0.00 0.05 0.13  0.09
 [8,] 0.13 0.14 0.15 0.15 0.13 0.13 0.05 0.00 0.16  0.13
 [9,] 0.16 0.16 0.11 0.11 0.09 0.09 0.13 0.16 0.00  0.07
[10,] 0.14 0.15 0.09 0.09 0.06 0.06 0.09 0.13 0.07  0.00
```

The above is the distance matrix we obtained from third distancefunction.



The above is the plot without nodes shown in it. We plotted this tree so that we canvisualize better.

Interpretation:

Sequence 6 and Sequence 5 have evolved from common ancestor (Node 15).Sequence 9 and Sequence 10 have evolved from common ancestor (Node13)

Sequence 1 and Node 2 have evolved from common ancestor (Node 18).

Sequence 7 and Node 8 have evolved from common ancestor (Node 17).

Sequence 4 and Node 3 have evolved from common ancestor (Node16)

Node 15 and node 13 have common ancestor from node 12.

Node 17 and node 18 have common ancestor from node 14

Node 12 and node 16 have common ancestor from node 11

All the evolutions started at Node 11.

Question 6)

By assuming every sequence is a Markov chain with state space {A,C,G,T} and initial distribution $(X = a) = 1/4$, $a \in$ {A,C,G,T}. Obtain the estimates of one step transition probability matrix.Are these Markov chains ergodic? Justify your answer.

Solution) Markov Chain:

A Markov process is a stochastic process that satisfies the Markov property. In simpler terms, it is a process for which predictions can be made regarding future outcomes based solely on its present state and most importantly such predictions are just as good as the ones that could be made knowing the process's full history.

Markov Property:
A sequence of random variable $\{X_n, n \geq 0\}$ with state space S and $x_0, x_1, \ldots, x_n \in S$ is said to follow Markov property if $P(X_{n+1} = x_{n+1} | X_0 = x_0, X_1 = x_1, \ldots, X_n = x_n) = P(X_{n+1} = x_{n+1} | X_n = x_n)$ a.s.

Markov Chain:
In a general sense, a Markov process having a discrete index set is known as the Markov chain.
Or
A stochastic process is said to be the Markov chain if it has a discrete index set and state space as well as it follows Markov property.
For DNA sequences state space is {A,C,G,T} which is finite.

Transition Probability Matrix (tpm)
It is a matrix of one step transition probabilities with row sums equal to zero.

Ergodicity
A state is said to be ergodic if it is aperiodic and non-null persistent.

A i state has period k if any return to state i must occur in multiples of k time steps. Formally, theperiod of a state is defined as $k = \{n: P(X_n = i | X_0 = i) > 0\}$
If k is equal to one we say state i is aperiodic.

R code :

```
> rm(list=ls())
> library('seqinr')
> library(markovchain)
> D=read.fasta(file.choose(),seqtype ="DNA")
> s1=D$AD90231[1:271]
> s2=D$AD00663[1:271]
> s3=D$CX63709[1:271]
> s4=D$CY08214[1:271]
> s5=D$DAF142766[1:271]
> s6=D$DAF142769[1:271]
> s7=D$FZ77158[1:271]
> s8=D$FZ77160[1:271]
> s9=D$KAF365891[1:271]
> s10=D$MEU069917[1:271]
> S=list(s1,s2,s3,s4,s5,s6,s7,s8,s9,s10)
> View(S)
>
```

We imported the data. We decided to use the package *markovchain* for obtaining the one step tpm for all the sequences, and to check if those were ergodic in nature.

SEQUENCE 1:

```
 mf1=markovchainFit(data=s1)$estimate;mf1MLE Fit

 A  4 - dimensional discrete Markov Chain defined by the

following states:

 a, c, g, t

 The transition matrix  (by rows)  is defined as follows:

 a         c         g         t

 a 0.2571429 0.2428571 0.1714286 0.3285714

 c 0.2105263 0.2763158 0.1710526 0.3421053

 g 0.2857143 0.2448980 0.2040816 0.2653061

 t 0.2800000 0.3466667 0.2000000 0.1733333
```

The above is the tpm for sequence 1.

```
  is.irreducible(mf1)

[1] TRUE

  period(mf1)

[1] 1
```

Interpretation:

As we can see none of the values of tpm are less than or equal to zero. The state space i.e. {A,C,G,T} is finite.

The function *is.irreducible()* shows that the tpm is irreducible. Hence the states are Non-null persistent.

The *period()* function shows us that the period is 1. Also the tpm has all non-zero entries which indicates that all states communicate with each other. Hence the period for all states is 1.

Since all states are Non null persistent and have period 1, hence all states are ergodic in nature.

SEQUENCE 2:

```
  mf2=markovchainFit(data=s2)$estimate;mf2MLE Fit

  A  4 - dimensional discrete Markov Chain defined by the

following states:

  a, c, g, t

  The transition matrix  (by rows)  is defined as follows:

  a         c         g         t

  a 0.2676056 0.2535211 0.1690141 0.3098592

  c 0.2105263 0.2763158 0.1710526 0.3421053

  g 0.2857143 0.2448980 0.2040816 0.2653061

  t 0.2837838 0.3378378 0.2027027 0.1756757
```

The above is the tpm for sequence 2.

```
  is.irreducible(mf2)

[1] TRUE

  period(mf2)

[1] 1
```

Interpretation:

As we can see none of the values of tpm are less than or equal to zero. The state space i.e. {A,C,G,T} is finite.

The function *is.irreducible()* shows that the tpm is irreducible. Hence the states are Non-null persistent.

The *period()* function shows us that the period is 1. Also the tpm has all non-zero entries which indicates that all states communicate with each other. Hence the period for all states is 1.

Since all states are Non null persistent and have period 1, hence all states are ergodic in nature.

SEQUENCE 3:

```
mf3=markovchainFit(data=s3)$estimate;mf3MLE Fit

  A  4 - dimensional discrete Markov Chain defined by the

 following states:

  a, c, g, t

  The transition matrix  (by rows)  is defined as follows:

  a        c        g        t

  a 0.2142857 0.2714286 0.2000000 0.3142857

  c 0.2361111 0.2500000 0.2083333 0.3055556

  g 0.3269231 0.1730769 0.1923077 0.3076923

  t 0.2631579 0.3421053 0.1842105 0.2105263
```

The above is the tpm for sequence 3.

```
  is.irreducible(mf3)
[1] TRUE

period(mf3)

[1] 1
```

Interpretation:
As we can see none of the values of tpm are less than or equal to zero. The state space i.e. {A,C,G,T} is finite.

The function *is.irreducible()* shows that the tpm is irreducible. Hence the states are Non-null persistent.

The *period()* function shows us that the period is 1. Also the tpm has all non-zero entries which indicates that all states communicate with each other. Hence the period for all states is 1.

Since all states are Non null persistent and have period 1, hence all states are ergodic in nature.

SEQUENCE 4:

```
  mf4=markovchainFit(data=s4)$estimate;mf4MLE Fit

  A  4 - dimensional discrete Markov Chain defined by the

following states:

  a, c, g, t

  The transition matrix  (by rows)  is defined as follows:

  a         c         g         t

  a 0.2253521 0.2535211 0.1971831 0.3239437

  c 0.2352941 0.2058824 0.1911765 0.3676471

  g 0.3529412 0.1764706 0.1960784 0.2745098

  t 0.2500000 0.3375000 0.1875000 0.2250000
```

The above is the tpm for sequence 4.

```
  is.irreducible(mf4)

[1] TRUE

  period(mf4)

[1] 1
```

Interpretation:

As we can see none of the values of tpm are less than or equal to zero. The state space i.e. {A,C,G,T} is finite.

The function *is.irreducible()* shows that the tpm is irreducible. Hence the states are Non-null persistent.

The *period()* function shows us that the period is 1. Also the tpm has all non-zero entries which indicates that all states communicate with each other. Hence the period for all states is 1.

Since all states are Non null persistent and have period 1, hence all states are ergodic in nature.

SEQUENCE 5:

```
  mf5=markovchainFit(data=s5)$estimate;mf5MLE Fit

  A  4 - dimensional discrete Markov Chain defined by the

following states:

  a, c, g, t

  The transition matrix  (by rows)  is defined as follows:

  a         c         g         t

  a 0.2500000 0.2647059 0.1470588 0.3382353

  c 0.2112676 0.2253521 0.1549296 0.4084507

  g 0.3333333 0.1875000 0.1875000 0.2916667

  t 0.2289157 0.3373494 0.2289157 0.2048193
```

The above is the tpm for sequence 5.

```
  is.irreducible(mf5)

[1] TRUE

  period(mf5)

[1] 1
```

Interpretation:

As we can see none of the values of tpm are less than or equal to zero. The state space i.e. {A,C,G,T} is finite.

The function *is.irreducible()* shows that the tpm is irreducible. Hence the states are Non-null persistent.

The *period()* function shows us that the period is 1. Also the tpm has all non-zero entries which indicates that all states communicate with each other. Hence the period for all states is 1.

Since all states are Non null persistent and have period 1, hence all states are ergodic in nature.

SEQUENCE 6:

```
  mf6=markovchainFit(data=s6)$estimate;mf6MLE Fit

  A  4 - dimensional discrete Markov Chain defined by the

following states:

  a, c, g, t

  The transition matrix  (by rows)  is defined as follows:

  a         c         g         t

  a 0.2676056 0.2253521 0.1549296 0.3521127

  c 0.2173913 0.2463768 0.1594203 0.3768116

  g 0.3541667 0.1875000 0.1666667 0.2916667

  t 0.2317073 0.3292683 0.2317073 0.2073171
```

The above is the tpm for sequence 6.

```
  is.irreducible(mf6)

[1] TRUE

  period(mf6)

[1] 1
```

Interpretation:

As we can see none of the values of tpm are less than or equal to zero. The state space i.e. {A,C,G,T} is finite.

The function *is.irreducible()* shows that the tpm is irreducible. Hence the states are Non-null persistent.

The *period()* function shows us that the period is 1. Also the tpm has all non-zero entries which indicates that all states communicate with each other. Hence the period for all states is 1.

Since all states are Non null persistent and have period 1, hence all states are ergodic in nature.

SEQUENCE 7:

```
  mf7=markovchainFit(data=s7)$estimate;mf7MLE Fit

  A  4 - dimensional discrete Markov Chain defined by the

following states:

  a, c, g, t

  The transition matrix  (by rows)  is defined as follows:

  a         c         g         t

  a 0.2571429 0.2714286 0.1571429 0.3142857

  c 0.2837838 0.2702703 0.1216216 0.3243243

  g 0.3469388 0.1428571 0.2448980 0.2653061

  t 0.1688312 0.3636364 0.2337662 0.2337662
```

The above is the tpm for sequence 7.

```
  is.irreducible(mf7)

[1] TRUE

  period(mf7)

[1] 1
```

Interpretation:

As we can see none of the values of tpm are less than or equal to zero. The state space i.e. {A,C,G,T} is finite.

The function *is.irreducible()* shows that the tpm is irreducible. Hence the states are Non-null persistent.

The *period()* function shows us that the period is 1. Also the tpm has all non-zero entries which indicates that all states communicate with each other. Hence the period for all states is 1.

Since all states are Non null persistent and have period 1, hence all states are ergodic in nature.

SEQUENCE 8:

```
mf8=markovchainFit(data=s8)$estimate;mf8MLE Fit

A  4 - dimensional discrete Markov Chain defined by the

following states:

a, c, g, t

The transition matrix  (by rows)  is defined as follows:

a         c         g         t

a 0.2500000 0.2647059 0.1617647 0.3235294

c 0.2692308 0.2564103 0.1282051 0.3461538

g 0.3018868 0.1886792 0.2830189 0.2264151

t 0.1830986 0.4225352 0.2535211 0.1408451
```

The above is the tpm for sequence 8.

```
is.irreducible(mf8)

[1] TRUE

period(mf8)

[1] 1
```

Interpretation:

As we can see none of the values of tpm are less than or equal to zero. The state space i.e. {A,C,G,T} is finite.

The function *is.irreducible()* shows that the tpm is irreducible. Hence the states are Non-null persistent.

The *period()* function shows us that the period is 1. Also the tpm has all non-zero entries which indicates that all states communicate with each other. Hence the period for all states is 1.

Since all states are Non null persistent and have period 1, hence all states are ergodic in nature.

SEQUENCE 9:

```
  mf9=markovchainFit(data=s9)$estimate;mf9MLE Fit

  A  4 - dimensional discrete Markov Chain defined by the

following states:

  a, c, g, t

  The transition matrix  (by rows)  is defined as follows:

  a         c         g         t

  a 0.2500000 0.2941176 0.1764706 0.2794118

  c 0.2692308 0.2564103 0.1538462 0.3205128

  g 0.2916667 0.2500000 0.1875000 0.2708333

  t 0.1973684 0.3421053 0.2105263 0.2500000
```

The above is the tpm for sequence 9.

```
  is.irreducible(mf9)

[1] TRUE

  period(mf9)

[1] 1
```

Interpretation:

As we can see none of the values of tpm are less than or equal to zero. The state space i.e. {A,C,G,T} is finite.

The function *is.irreducible()* shows that the tpm is irreducible. Hence the states are Non-null persistent.

The *period()* function shows us that the period is 1. Also the tpm has all non-zero entries which indicates that all states communicate with each other. Hence the period for all states is 1.

Since all states are Non null persistent and have period 1, hence all states are ergodic in nature.

SEQUENCE 10:

```
  mf10=markovchainFit(data=s10)$estimate;mf10MLE Fit

  A  4 - dimensional discrete Markov Chain defined by the following

states:

  a, c, g, t

  The transition matrix  (by rows)  is defined as follows:

  a         c         g         t

  a 0.2608696 0.2463768 0.1594203 0.3333333

  c 0.2285714 0.2428571 0.1428571 0.3857143

  g 0.3043478 0.2173913 0.1956522 0.2826087

  t 0.2352941 0.3058824 0.2000000 0.2588235
```

The above is the tpm for sequence 10.

```
  is.irreducible(mf10)

[1] TRUE

  period(mf10)

[1] 1
```

Interpretation:

As we can see none of the values of tpm are less than or equal to zero. The state space i.e. {A,C,G,T} is finite.

The function *is.irreducible()* shows that the tpm is irreducible. Hence the states are Non-null persistent.

The *period()* function shows us that the period is 1. Also the tpm has all non-zero entries which indicates that all states communicate with each other. Hence the period for all states is 1.

Since all states are Non null persistent and have period 1, hence all states are ergodic in nature.