# Assignment 5

Jieran Sun, Hui Jeong (HJ) Jung, Gudmundur Björgvin Magnusson

2023-03-28

## Question 12

**(1)**

$$P(t + dt) = P(dt)P(t) = (I + Rdt)P(t) \tag{1}$$

Therefore we can show that

$$\frac{dP(t)}{dt} = RP(t) \tag{2}$$

**(2)**

**TODO**

## Question 13

**(1) The joint probability of the tree is**

**TODO : not sure what the "long version" is supposed to be, bc the one below is the "simple" one**

$$P(X, Z|T) = P(Z_4) * P(X_5|Z_4) * P(Z_3|Z_4) * P(Z_2|Z_3) * P(Z_1|Z_3) \tag{3}$$
$$* P(X_4|Z_2) * P(X_3|Z_2) * P(X_2|Z_1) * P(X_1|Z_1) \tag{4}$$

**(2) To do the naive calculation of P(X|T) via brute-force marginalization over the hidden nodes Z,**

for each node X, we have to marginalize out all the internal nodes Z that it is dependent on. Here is an example with X4.

$$P(X_4|T) = \sum_Z P(X_4, Z|T) = \sum_Z P(X_4|Z_2)P(Z_2|Z_3)P(Z_3|Z_4)P(Z_4) \tag{5}$$

$$= \sum_Z P(X_4, Z_2, Z_3, Z_4) \tag{6}$$

Thus for X4, we would have to sum over 3 times. For X1, X2, X3, and X4 they all depend on 3 Zs so they also have to sum 3 times. X5 only depends on Z4, therefore only has to sum once. In total the brute force method would require 13 summations.

**(3)**

$$P(X, Z|T) = P(Z_4) * P(X_5|Z_4) * P(Z_3|Z_4) * P(Z_2|Z_3) * P(Z_1|Z_3) \tag{7}$$
$$* P(X_4|Z_2) * P(X_3|Z_2) * P(X_2|Z_1) * P(X_1|Z_1) \tag{8}$$

By rearranging the expression in such way we only need to do 4 summations.

## Question 14

**(1)**

```
if(!require("phangorn")) {
  install.packages("phangorn")
}
```

```
## Loading required package: phangorn
```

```
## Loading required package: ape
```
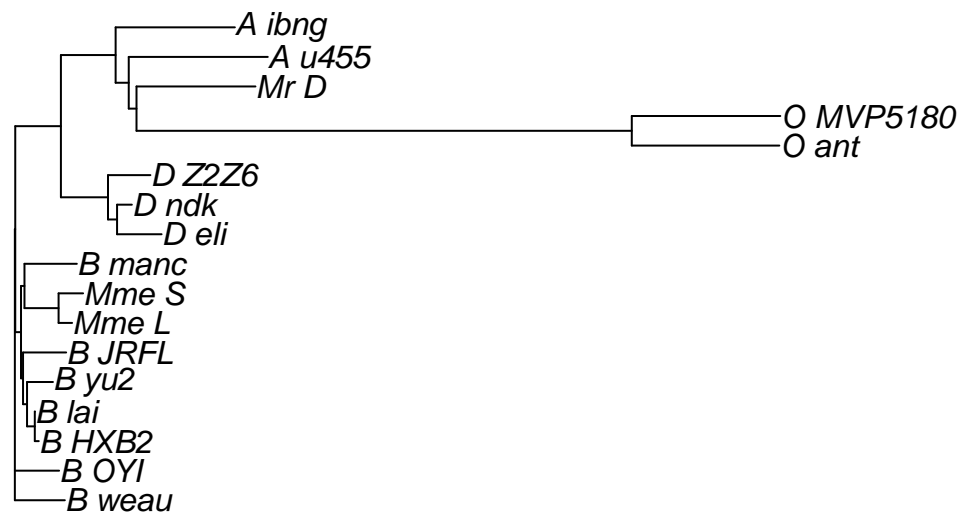
```
if(!require("ape")) {
  install.packages("ape")
}

library(phangorn)
library(ape)
```

```
ParisRT <- read.dna("ParisRT.txt")
```

**(2)**

```
distParis <- dist.dna(ParisRT)
initTree <- NJ(distParis)
plot(initTree)
```

### (3)

```
kimura <- pml(tree= initTree, data= phyDat(ParisRT), model= "K80")
kimura$logLik
```

```
## [1] -3003.487
```

The log likelihood of the fitted model is -3003.487.

**(4)**

The values of the optimised rate matrix can be found below.

```
optimParam <- optim.pml(kimura, optQ= TRUE)
```

```
## optimize edge weights:   -3003.487 --> -2992.981
## optimize rate matrix:   -2992.981 --> -2863.264
## optimize edge weights:   -2863.264 --> -2862.477
## optimize rate matrix:   -2862.477 --> -2862.456
## optimize edge weights:   -2862.456 --> -2862.455
## optimize rate matrix:   -2862.455 --> -2862.455
## optimize edge weights:   -2862.455 --> -2862.455
## optimize rate matrix:   -2862.455 --> -2862.455
## optimize edge weights:   -2862.455 --> -2862.455
```

```
optimParam
```

```
## model: K80
## loglikelihood: -2862.455
## unconstrained loglikelihood: -2098.897
##
## Rate matrix:
##          a         c         g         t
## a 0.000000 2.4318480 6.8571651 1.118323
## c 2.431848 0.0000000 0.6119506 7.262319
## g 6.857165 0.6119506 0.0000000 1.000000
## t 1.118323 7.2623187 1.0000000 0.000000
##
## Base frequencies:
##    a    c    g    t
## 0.25 0.25 0.25 0.25
```

**(5)**

After optimizing with respect to branch lengths, nucleotide substitution rates, and tree topology, the results are as below.

```
optimParam2 <- optim.pml(kimura, optQ = TRUE, optNni = TRUE, optEdge = TRUE)
```

```
## optimize edge weights:  -3003.487 --> -2992.981
## optimize rate matrix:  -2992.981 --> -2863.264
## optimize edge weights:  -2863.264 --> -2862.477
## optimize topology:  -2862.477 --> -2849.886  NNI moves:  5
## optimize rate matrix:  -2849.886 --> -2849.791
## optimize edge weights:  -2849.791 --> -2849.789
## optimize topology:  -2849.789 --> -2849.789  NNI moves:  0
## optimize rate matrix:  -2849.789 --> -2849.789
## optimize edge weights:  -2849.789 --> -2849.789
## optimize rate matrix:  -2849.789 --> -2849.789
## optimize edge weights:  -2849.789 --> -2849.789
```

```
optimParam2$logLik
```

```
## [1] -2849.789
```

The log likelihood is -2849.789.

**(6)**

```
bootPML <- bootstrap.pml(optimParam2, optNni= TRUE)
```

In this bootstrapping function, we are resampling the sequences that we are using to construct a phylogenetic tree, and seeing if the same branch is observed even when generating a new tree based on bootstrapped data, which would indicate confidence in the observed branch.

**(7)**

```
plotBS(tree= optimParam2$tree, BStrees = bootPML, type= "phylogram")
```
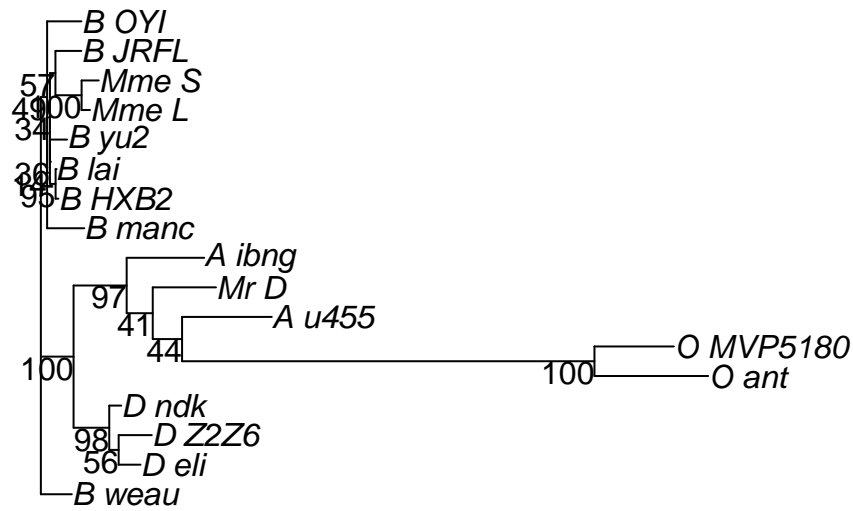


Judging from this plot, it is more likely that Mme_S was more likely to have infected the patient Mme_L.