SMCB : Projet 5

Marie-Claire Imdilemitsch, Anna Taitze, Ieva Pudziuvelyte, Sophia Houkamdi

• Problem 12:

1). $\forall t > 0$, $P(t+dt) = P(dt)\,P(t)$ (Chapman-Kolmogorov equation)

$= \left(\underbrace{P(0)}_{I} + R\,dt\right) P(t)$

$= P(t) + R\,P(t)\,dt$

So: $\dfrac{dP(t)}{dt} = \dfrac{P(t+dt) - P(t)}{dt} = R\,P(t)$

2) $\forall t > 0$: $P(t)\,\vec{\pi} = \vec{\pi}$

So $\vec{\pi} = P(dt)\,\vec{\pi} = (I + R\,dt)\,\vec{\pi} = \vec{\pi} + R\vec{\pi}\,dt$

Then: $R\vec{\pi}\,dt = 0$

$R\vec{\pi} = 0$

• Problem 13:

1). The joint probability $\mathbb{P}(X, Z \mid T)$ is given by:

$$\mathbb{P}(x_1, x_2, x_3, x_4, x_5, z_1, z_2, z_3, z_4) = \mathbb{P}(z_4)\cdot\mathbb{P}(z_3 \mid z_4)\cdot\mathbb{P}(z_2 \mid z_3)\cdot\mathbb{P}(z_1 \mid z_3)\times \mathbb{P}(x_5 \mid z_4)\cdot\mathbb{P}(x_4 \mid z_2)\cdot\mathbb{P}(x_3 \mid z_2)\cdot\mathbb{P}(x_2 \mid z_1)\,\mathbb{P}(x_1 \mid z_1)$$

2). ~~Each~~ Four of the hidden states can take 4 values $\{A, C, G, T\}$, so the naive calculation of $P(X \mid T)$ via brute-force marginalization over the hidden nodes Z has $4^4 = 256$ summation steps.

3). $A = \sum_{z_1}\sum_{z_2}\sum_{z_3}\sum_{z_4} \mathbb{P}(x_1, \dots, x_5, z_1, \dots, z_5) = \sum_{z_1}\sum_{z_2}\sum_{z_3}\sum_{z_4} \mathbb{P}(z_4)\cdot\mathbb{P}(z_3 \mid z_4)\,\mathbb{P}(z_2 \mid z_3)\cdot\mathbb{P}(z_1 \mid z_3) \times \mathbb{P}(x_5 \mid z_4)\cdot\mathbb{P}(x_4 \mid z_2)\,\mathbb{P}(x_3 \mid z_2)\cdot \times \mathbb{P}(x_2 \mid z_1)\;\mathbb{P}(x_1 \mid z_1)$

$$= \sum_{z_4} \mathbb{P}(z_4)\,\mathbb{P}(x_5 \mid z_4)\left[\left[\sum_{z_3}\left[\mathbb{P}(z_3 \mid z_4)\left(\sum_{z_2}\mathbb{P}(z_2 \mid z_3)\,\mathbb{P}(x_4 \mid z_2)\,\mathbb{P}(x_3 \mid z_2)\right)\times\left(\sum_{z_1}\mathbb{P}(z_1 \mid z_3)\,\mathbb{P}(x_2 \mid z_1)\,\mathbb{P}(x_1 \mid z_1)\right)\right]\right]\right]$$

We can thus compute this sum in a recursive manner:

- $\beta(z_1) = \mathbb{P}(x_2 \mid z_1)\,\mathbb{P}(x_1 \mid z_1)$ — 1 sum, for 4 different values z_1
- $\beta(z_2) = \mathbb{P}(x_4 \mid z_2)\,\mathbb{P}(x_3 \mid z_2)$ — 1 sum, for 4 different values z_2
- $\alpha(z_3) = \sum_{z_2}\mathbb{P}(z_2 \mid z_3)\,\beta(z_2)\sum_{z_1}\mathbb{P}(z_1 \mid z_3)\,\beta(z_1) = \sum_{z_1, z_2}\mathbb{P}(z_2 \mid z_3)\,\mathbb{P}(z_1 \mid z_3)\,\beta(z_2)\,\beta(z_1)$ — 4×4 sums (16 sums) for 4 different values z_3
- $\alpha(z_4) = \sum_{z_3}\mathbb{P}(z_3 \mid z_4)\,\beta(z_3)$ — 4 sums, for 4 different values of z_4
- $A = \sum_{z_4}\mathbb{P}(z_4)\cdot\mathbb{P}(x_5 \mid z_4)\,\alpha(z_4)$ — 4 sums, once

- Each of the four values of $o(\mathbb{Z}_3)$ requires 4^2 terms, so $4 \times 4^2 = 64$ summations are required to know all values of $o(\mathbb{Z}_3)$
- Each of the four values of $o(\mathbb{Z}_4)$ requires 4 terms, so $4 \times 4 = 16$ summations are required to know all the values of $o(\mathbb{Z}_4)$ (once the values of $o(\mathbb{Z}_3)$ are determined.
- To calculate A, we need to do 4 summations.

Overall, $64 + 16 + 4 = 84$ summations are required.

Project_5

Contents

Problem 14

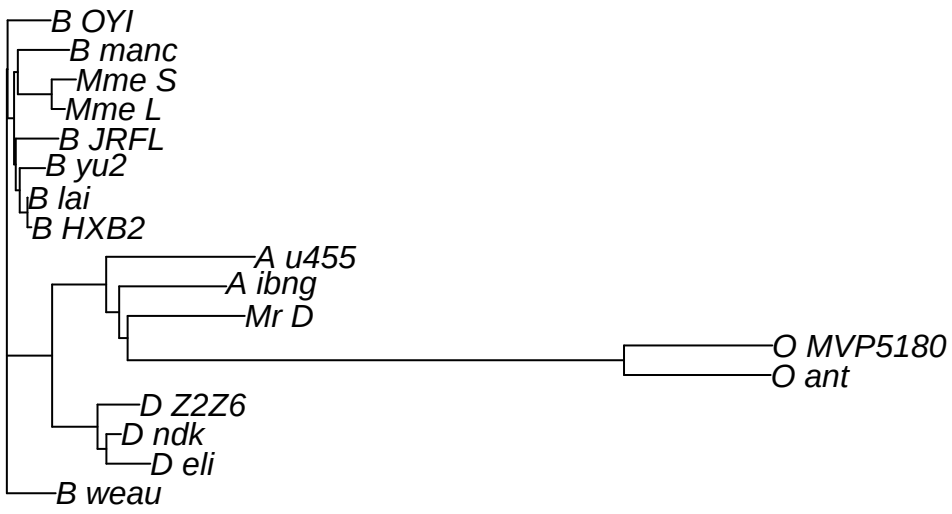
1. Install and load the R packages phangorn and ape. Load the alignment ParisRT.txt into memory using the function read.dna().

```
#Load packages
library(ggplot2)
library(phangorn)
library(ape)

#Load the data
paris_rt <- read.dna("ParisRT.txt")
```

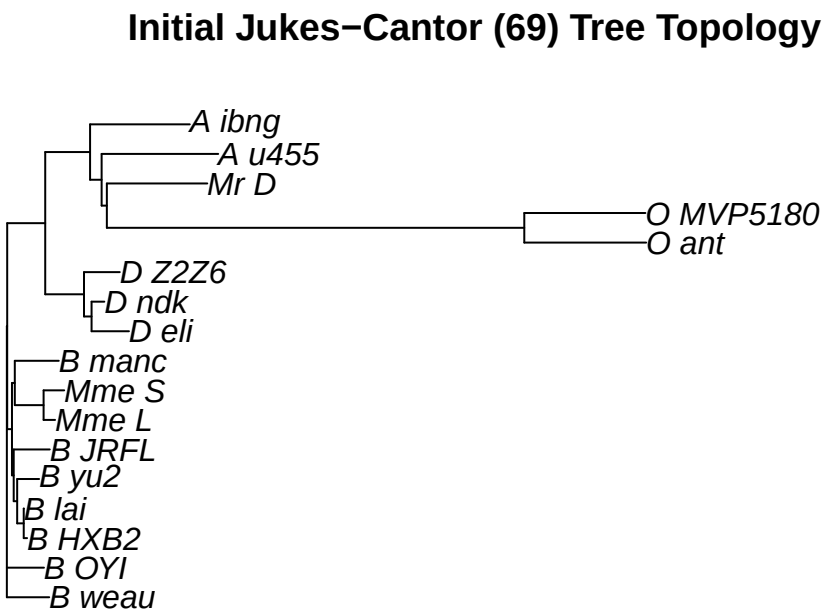
2. Create an initial tree topology for the alignment, using neighbour joining with the function NJ(). Base this on pairwise distances between sequences under the a. Kimura (1980), b.(Tamura and Nei 1993) and c. (Jukes and Cantor 1969) nucleotide substitution models, computed using the function dist.dna(). Plot the initial trees.

```
#Calculate the pairwise distances under each substitution model:
  #Kimura (1980)
dist_kimura <- dist.dna(paris_rt, model = "K80")
  #Tamura-Nei (1993)
dist_tamura_nei <- dist.dna(paris_rt, model = "TN93")
  #Jukes-Cantor (1969)
dist_jukes_cantor <- dist.dna(paris_rt, model = "JC69")


#Create initial trees using the neighbor-joining method
tree_K80 <- NJ(dist_kimura)
tree_TN93 <- NJ(dist_tamura_nei)
tree_JC69 <- NJ(dist_jukes_cantor)


#Plot the trees
  #Kimura (1980)
plot(tree_K80, main = "Initial Kimoura (80) Tree Topology")
```


```
  #Tamura-Nei (1993)
plot(tree_TN93, main = "Initial Tamura-Nei (93) Tree Topology")
```




```
  #Jukes-Cantor (1969)
plot(tree_JC69, main = "Initial Jukes-Cantor (69) Tree Topology")
```



3. Use the function pml() to fit the Kimura model (model = "K80") to the above tree (a) and the alignment. Note that the function expects data = phyDat(alignment). What is the log likelihood of the fitted model?

```
#Transform alignment into the appropriate format for pml
align_phydat <- phyDat(paris_rt)

#Fit the parameters of K80
paris_rt_k80_param <- pml(tree = tree_K80, data = align_phydat, model = "K80")
print(paste("The log-likelihood is", paris_rt_k80_param$logLik))
```

```
## [1] "The log-likelihood is -3003.48688257944"
```

4.The function optim.pml() can be used to optimise parameters of a phylogenetic model. Find the optimal parameters of the Kimura (1980) nucleotide substitution model whilst the other parameters are held fixed. What are the values in the optimised rate matrix?

```
#Optimize the Kimura parameters with others held fixed
paris_rt_k80_param_opt <- optim.pml(paris_rt_k80_param, model = "K80", optQ = TRUE, optEdge = FALSE)
```

```
## optimize rate matrix:  -3003.487 --> -2884.408
## optimize rate matrix:  -2884.408 --> -2884.408
```

```
#summary(paris_rt_k80_param_opt)

#Print values of optimized rate matrix
rate_matrix <- paris_rt_k80_param_opt$Q
print(paste("The values in the optimised rate matrix of the Kimura model are alpha (transitions rate)="
```

```
## [1] "The values in the optimised rate matrix of the Kimura model are alpha (transitions rate)= 4.976
```

5. Optimise the Kimura model with respect to branch lengths, nucleotide substitution rates, and tree topology simultaneously. What is the log likelihood of the optimised model?

```
print(paste("The new log-likelihood is", paris_rt_k80_param_opt$logLik))
```

```
## [1] "The new log-likelihood is -2884.407689035"
```

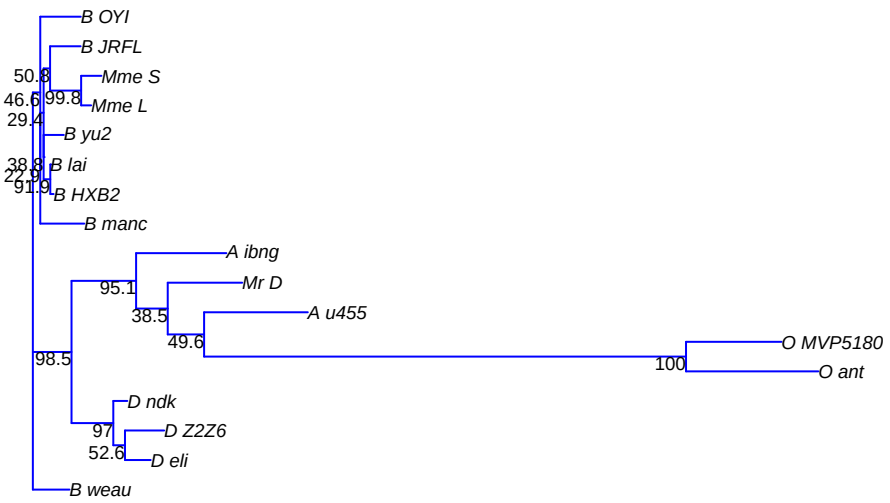
6. The function bootstrap.pml() fits phylogenetic models to bootstrap resamples of the data. Run it on the optimised model from step 5, but pass the argument optNni = TRUE to allow for a different topology for each bootstrap run. What, exactly, is being resampled?

```
# Running 1000 bootstrap samples
bootstrap_results <- bootstrap.pml(paris_rt_k80_param_opt_all, bs = 1000, model = "K80", optNni = TRUE)
```

The data being resampled with replacement here are the columns (positions of the alignments of the sequence data). Each pseudo-replicate is the same size as the original dataset but may include some columns multiple times and omit others.

7.Use plotBS() with type = "phylogram" to plot the optimised tree (from step 5) with the bootstrap support on the edges. Which nurse ("Mme S" or "Mr D") is more likely to have infected the patient "Mme L"?

```
plotBS(tree = paris_rt_k80_param_opt_all$tree, BStrees = bootstrap_results, type = "phylogram",
       cex = 0.6, cex.tip = 0.6, lwd = 2, edge.color = "blue", use.edge.length = TRUE,
       main = "Phylogenetic Tree K80 model with Bootstrap Values")
```



From the tree above, we observe that Mme S and Mme L group more closely together, with high bootstrap support (99.8). We can thus conclude that the nurse Mme S is more likely to have infected the patient Mme L.

```
library(rmarkdown)
render("project5.Rmd", pdf_document(TRUE), "Project5_Indilewitsch_Toidze_Houhamdi_Pudziuvelyte.pdf")
```