Project 5

Santiago Castro Dau, June Monge, Rachita Kumar, Sarah Lötscher

2022-03-31

Problem 12

1. Show that: $\frac{d\mathbf{P}(t)}{dt} = \mathbf{R}\mathbf{P}(t)$

Using Chapman - Kolmogorov equation,

$$\begin{aligned}
\mathbf{P}(dt+t) &= \mathbf{P}(dt)\mathbf{P}(t) \\
&= (\mathbf{I} + \mathbf{R}dt)\mathbf{P}(t) \\
&= \mathbf{P}(t) + \mathbf{R}\mathbf{P}(t)dt \\
\frac{\mathbf{P}(t+dt) - \mathbf{P}(t)}{dt} &= \mathbf{R}\mathbf{P}(t) \\
\implies \frac{dP(t)}{dt} &= \mathbf{R}\mathbf{P}(t)
\end{aligned}$$

Hence Proved.

2. Show that $R\pi = 0$**.**

We know that for the stationary distribution π the following holds, $P(t)\pi = \pi$. Since we are dealing with convergence, the previous equation must hold for any time after t, so the following must also hold, $P(t+dt)\pi = \pi$. By way of the Chapman - Kolmogorov equation we can rewrite the previous equation as $P(dt+t)\pi = (I + Rdt)P(t)\pi$. Thus $P(dt+t)\pi = P(t)\pi + RdtP(t)\pi$. Since $P(t)\pi = \pi$, $P(dt+t)\pi = \pi + Rdt\pi$ and therefore $Rdt\pi$ must be 0. Since dt is just a constant we can say that $R\pi = 0$.

Problem 13

1. What is the joint probability $P(X, Z|T)$ **of the tree?**

Ommiting the conventionality on the tree T for clearness.

$$P(X, Z|T) = P(Z_4)P(Z_3|Z_4)P(X_5|Z_4)P(Z_2|Z_3)P(Z_1|Z_3)P(X_4|Z_2)P(X_3|Z_2)P(X_2|Z_1)P(X_1|Z_1)$$

2. How many summation steps would be required for the naive calculation of $P(X|T)$ **via brute-force marginalization over the hidden nodes Z?**

The number of summation steps for the calculation of $P(X|T)$ would be: $4 * 4 * 4 * 4 = 256$

3. Rearrange the expression $P(X|T)$ such that the number of operations is minimized. How many summation steps are required now for the calculation of $P(X|T)$?

$$
\begin{aligned}
P(X|T) &= \sum_{Z_4}\sum_{Z_3}\sum_{Z_2}\sum_{Z_1} P(X,Z|T) \\
&= \sum_{Z_4}\sum_{Z_3}\sum_{Z_2}\sum_{Z_1} P(Z_4)P(Z_3|Z_4)P(X_5|Z_4)P(Z_2|Z_3)P(Z_1|Z_3)P(X_4|Z_2)P(X_3|Z_2)P(X_2|Z_1)P(X_1|Z_1) \\
&= \sum_{Z_4}\sum_{Z_3}\sum_{Z_2} P(Z_4)P(Z_3|Z_4)P(X_5|Z_4)P(Z_2|Z_3)P(X_4|Z_2)P(X_3|Z_2)\sum_{Z_1} P(Z_1|Z_3)P(X_2|Z_1)P(X_1|Z_1) \\
&= \sum_{Z_4}\sum_{Z_3} P(Z_4)P(Z_3|Z_4)P(X_5|Z_4)\sum_{Z_2} P(Z_2|Z_3)P(X_4|Z_2)P(X_3|Z_2)\sum_{Z_1} P(Z_1|Z_3)P(X_2|Z_1)P(X_1|Z_1) \\
&= \sum_{Z_4} P(Z_4)P(X_5|Z_4)\sum_{Z_3} P(Z_3|Z_4)\sum_{Z_2} P(Z_2|Z_3)P(X_4|Z_2)P(X_3|Z_2)\sum_{Z_1} P(Z_1|Z_3)P(X_2|Z_1)P(X_1|Z_1)
\end{aligned}
$$

Because the terms inside the summations concerning Z_1 and Z_2 only depend on Z_3 we only need to calculate them once per Z_3, Z_1 and Z_3, Z_2 value pairs $(4(4+4) = 32)$. Then we need to compute $P(Z_3|Z_4)$ 16 times, once for each per Z_3, Z_4 value pair. Finally we need to compute $P(Z_4)P(X_5|Z_4)$ 4 times, once for each value Z_4. In total we compute 52 terms. Then we sum them up by group and multiply them together to minimize the number of operations (3 multiplications and 52 summations).

Problem 14

```
## Warning: package 'phangorn' was built under R version 4.1.3
```

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

```
## Warning: package 'ape' was built under R version 4.1.3
```

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

```
alignment = 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 Kimura (1980) nucleotide substitution model, computed using the function dist.dna(). Plot the initial tree.

```
tree = NJ(dist.dna(alignment,model = "K80"))
plot.phylo(tree)
```

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

```
likelihood = pml(tree, data = phyDat(alignment), model ="K80")
likelihood
```

```
##
##  loglikelihood: -3003.487
##
## unconstrained loglikelihood: -2098.897
##
## Rate matrix:
##   a c g t
## a 0 1 1 1
## c 1 0 1 1
## g 1 1 0 1
## t 1 1 1 0
##
## Base frequencies:
## 0.25 0.25 0.25 0.25
```

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?

```
likelihood_optimised_rate_only =
  optim.pml(likelihood, optQ = TRUE, optEdge = FALSE, model='K80')
```

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

```
likelihood_optimised_rate_only
```

```
##
##  loglikelihood: -2884.408
##
## unconstrained loglikelihood: -2098.897
##
## Rate matrix:
##          a        c        g        t
## a 0.000000 1.000000 4.976955 1.000000
## c 1.000000 0.000000 1.000000 4.976955
## g 4.976955 1.000000 0.000000 1.000000
## t 1.000000 4.976955 1.000000 0.000000
##
## Base frequencies:
## 0.25 0.25 0.25 0.25
```

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?

```
likelihood_optimised =
  optim.pml(likelihood, optNni = TRUE, optEdge = TRUE, optQ = 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 --> -2854.512
##  optimize topology:  -2854.512 --> -2853.692
##  optimize topology:  -2853.692 --> -2849.886
##  optimize topology:  -2849.886 --> -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
```

```
likelihood_optimised
```

```
##
##  loglikelihood: -2849.789
##
## unconstrained loglikelihood: -2098.897
##
## Rate matrix:
##          a         c         g        t
## a 0.000000 2.4233712 7.1237289 1.148311
## c 2.423371 0.0000000 0.6224531 7.541752
## g 7.123729 0.6224531 0.0000000 1.000000
## t 1.148311 7.5417523 1.0000000 0.000000
##
## Base frequencies:
## 0.25 0.25 0.25 0.25
```

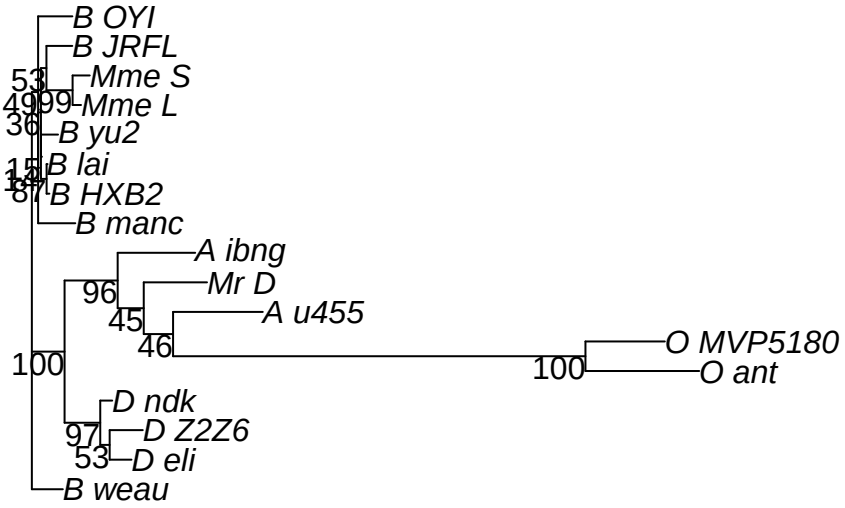
6. The function bootstrap.pml() fits phylogenetic models to bootstrap resamples of the data. Run it on the optimised model from step 5, but pass the argument optNni = TRUE to allow for a different topology for each bootstrap run. What, exactly, is being resampled?

The specific sites in the alignment are being re-sampled to generate bootstrap alignments in which we can re-run our analysis in order to get confidence estimates.

```
bootstrap = bootstrap.pml(x = likelihood_optimised, optNni = TRUE)
```

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(likelihood_optimised$tree, BStrees = bootstrap, type = "phylogram")
```



Mme S is more likely to have infected the patient Mme L. From the tree, we see that with a bootstrap support of ~99% the the samples from Mme S and Mme L have a more recent common ancestor than those of Mr D and Mme L.