Project 5

Richard Affolter

25 3 2021

Problem 12: Transition matrix, rate matrix, and stationary distribution

1. Show that $\frac{dP(t)}{dt} = RP(t)$

According to the Chapman-Kolmogorov equation we have $P(t+dt) = P(dt)P(t)$

$$\begin{aligned} P(t+dt) = P(dt+t) &= P(dt)P(t) \\ &= (P(0) + Rdt)P(t) \\ &= (I + Rdt)P(t) \\ &= P(t) + RdtP(t) \\ \Rightarrow \frac{P(t+dt) - P(t)}{dt} &= RP(t) \end{aligned}$$

And for infinitesimally small time interval dt, we have the definition of the derivative:

$$\lim_{dt \to 0} \frac{P(t+dt) - P(t)}{dt} = \frac{dP(t)}{dt}$$

2. Assume that the given Markov chain is ergodic with (unique) stationary distribution $\vec{\pi}$. Show that $\mathbf{R}\vec{\pi} = \vec{0}$.

From slide 13 we know that the stationary distribution is characterized by $P(w)\vec{\pi} = \vec{\pi}$. That means that if we set w to dt we get:

$$\begin{aligned} P(dt)\vec{\pi} &= \vec{\pi} \\ (I + Rdt)\vec{\pi} &= \vec{\pi} \\ \vec{\pi} + Rdt\vec{\pi} &= \vec{\pi} \\ \Rightarrow Rdt\vec{\pi} &= \vec{0} \\ \Rightarrow R\vec{\pi} &= \vec{0} \end{aligned}$$

Problem 13: Phylogenetic trees as Bayesian networks

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

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

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?

We have four internal nodes. For one site in the alignment a node can take on four different values (A,T,G or C). So for one site we have $4^4 = 256$ summation steps. For m sites we get $m \cdot 4^4 = 256 \cdot m$ summation steps.

If we denote the nucleotide of node Z_1 at site m as $Z_{1,j}$ then

$$P(X_1, X_2, X_3, X_4) =$$
$$\prod_{j=1}^{m} \sum_{Z_{4,i} \in \{T,C,A,G\}} \sum_{Z_{3,i} \in \{T,C,A,G\}} \sum_{Z_{2,i} \in \{T,C,A,G\}} \sum_{Z_{1,i} \in \{T,C,A,G\}} P(Z_{4,i})P(X_{5,i}|Z_{4,i})P(Z_{3,i}|Z_{4,i})$$
$$\times P(Z_{2,i}|Z_{3,i})P(X_{4,i}|Z_{2,i})P(X_{3,i}|Z_{2,i})$$
$$\times P(Z_{1,i}|Z_{3,i})P(X_{2,i}|Z_{1,i})P(X_{1,i}|Z_{1,i})$$

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 $P(X|T)$?

If we move the expression like this:

$$P(X_1, X_2, X_3, X_4) = \prod_{j=1}^{m} \sum_{Z_{4,i} \in \{T,C,A,G\}} P(Z_{4,i})P(X_{5,i}|Z_{4,i})$$
$$\times \sum_{Z_{3,i} \in \{T,C,A,G\}} P(Z_{3,i}|Z_{4,i})$$
$$\times \sum_{Z_{2,i} \in \{T,C,A,G\}} P(Z_{2,i}|Z_{3,i})P(X_{4,i}|Z_{2,i})P(X_{3,i}|Z_{2,i})$$
$$\times \sum_{Z_{1,i} \in \{T,C,A,G\}} P(Z_{1,i}|Z_{3,i})P(X_{2,i}|Z_{1,i})P(X_{1,i}|Z_{1,i})$$

We can move the summation down the tree. This means we only require $4 \cdot 4 \cdot m = 16 \cdot m$ summation steps.

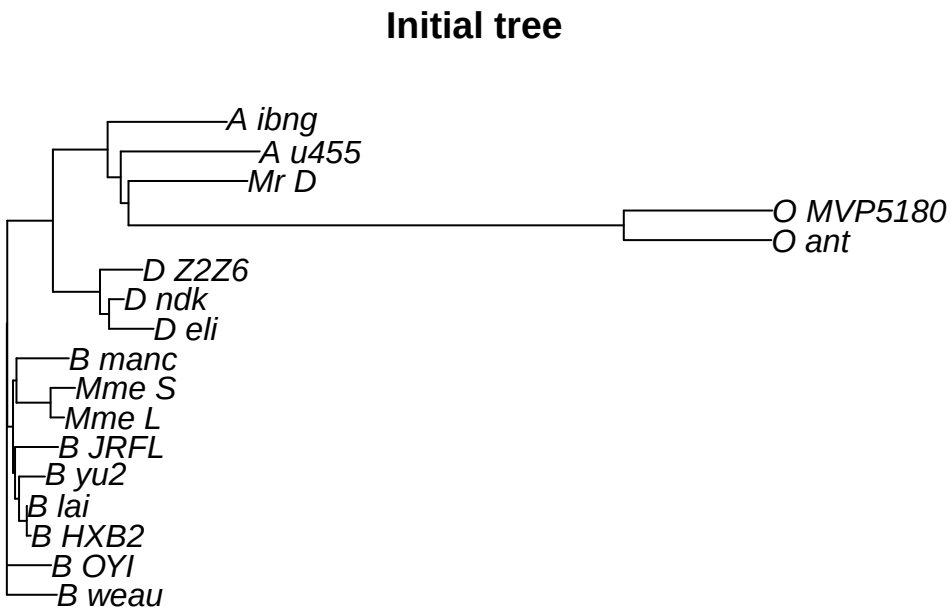
Problem 14: Learning phylogenetic trees from sequence alignment data

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

```
library(ape)

alignment = ape::read.dna("ParisRT.txt")

if(!require("phangorn")){
  packageurl <- "https://cran.r-project.org/src/contrib/Archive/phangorn/phangorn_2.5.5.tar.gz"
  install.packages(packageurl, repos=NULL, type="source")
}

library(phangorn)
```

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.

```
alignment_dist = dist.dna(x=alignment, model = "K80")
initial_tree = NJ(alignment_dist)
plot(initial_tree, main = "Initial 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?

```
kimura = pml(tree = initial_tree, model = "K80", data = phyDat(alignment))
cat("the log likelihood of the fitted model is: ", kimura$logLik)
```

```
## the log likelihood of the fitted model is:  -3003.487
```

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?

```
opt_sub = optim.pml(object = kimura, model = "K80",
          optNni = FALSE, optBf = FALSE, optQ = TRUE, optInv = FALSE,
          optGamma = FALSE, optEdge = FALSE, optRate = FALSE, optRooted = FALSE)
```

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

```
print(opt_sub)
```

```
##
##  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?

```
opt_sim = optim.pml(object = kimura, model = "K80",
          optNni = TRUE, optBf = FALSE, optQ = TRUE, optInv = FALSE,
          optGamma = FALSE, optEdge = TRUE, optRate = FALSE, optRooted = FALSE)
```

```
## optimize edge weights:  -3003.487 --> -2992.981
## optimize rate matrix:  -2992.981 --> -2873.703
## optimize edge weights:  -2873.703 --> -2872.892
## optimize topology:  -2872.892 --> -2864.379
## optimize topology:  -2864.379 --> -2863.544
## optimize topology:  -2863.544 --> -2859.775
## 5
## optimize rate matrix:  -2859.775 --> -2859.682
## optimize edge weights:  -2859.682 --> -2859.681
## optimize topology:  -2859.681 --> -2859.681
## 0
## optimize rate matrix:  -2859.681 --> -2859.681
## optimize edge weights:  -2859.681 --> -2859.681
```

```
print(opt_sim)
```

```
##
##  loglikelihood: -2859.681
##
## unconstrained loglikelihood: -2098.897
##
## Rate matrix:
##          a        c        g        t
## a 0.000000 1.000000 5.262145 1.000000
## c 1.000000 0.000000 1.000000 5.262145
## g 5.262145 1.000000 0.000000 1.000000
## t 1.000000 5.262145 1.000000 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?

```
set.seed(123)
bootstrapped = bootstrap.pml(x = opt_sim, optNni = TRUE,
                             control = pml.control(trace=0))
```

```
## Final p-score 404 after  1 nni operations
## Final p-score 361 after  2 nni operations
## Final p-score 452 after  1 nni operations
## Final p-score 399 after  2 nni operations
## Final p-score 369 after  0 nni operations
## Final p-score 420 after  0 nni operations
## Final p-score 401 after  0 nni operations
## Final p-score 436 after  0 nni operations
## Final p-score 398 after  1 nni operations
## Final p-score 451 after  0 nni operations
## Final p-score 399 after  2 nni operations
## Final p-score 393 after  1 nni operations
## Final p-score 407 after  1 nni operations
## Final p-score 427 after  1 nni operations
## Final p-score 425 after  1 nni operations
## Final p-score 404 after  0 nni operations
## Final p-score 390 after  0 nni operations
## Final p-score 437 after  1 nni operations
## Final p-score 413 after  0 nni operations
## Final p-score 376 after  3 nni operations
## Final p-score 397 after  0 nni operations
## Final p-score 422 after  2 nni operations
## Final p-score 407 after  0 nni operations
## Final p-score 417 after  2 nni operations
## Final p-score 435 after  1 nni operations
## Final p-score 388 after  0 nni operations
```

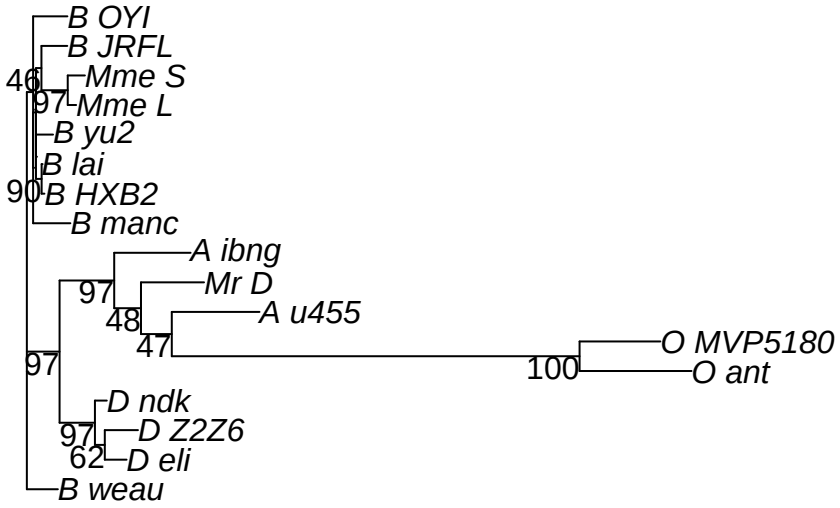
```
## Final p-score 392 after  0 nni operations
## Final p-score 436 after  2 nni operations
## Final p-score 394 after  0 nni operations
## Final p-score 345 after  0 nni operations
## Final p-score 432 after  0 nni operations
## Final p-score 394 after  0 nni operations
## Final p-score 376 after  1 nni operations
## Final p-score 405 after  0 nni operations
## Final p-score 439 after  1 nni operations
## Final p-score 411 after  1 nni operations
## Final p-score 400 after  0 nni operations
## Final p-score 379 after  0 nni operations
## Final p-score 414 after  0 nni operations
## Final p-score 384 after  4 nni operations
## Final p-score 399 after  4 nni operations
## Final p-score 404 after  1 nni operations
## Final p-score 383 after  0 nni operations
## Final p-score 369 after  0 nni operations
## Final p-score 362 after  0 nni operations
## Final p-score 442 after  1 nni operations
## Final p-score 390 after  0 nni operations
## Final p-score 431 after  0 nni operations
## Final p-score 386 after  0 nni operations
## Final p-score 383 after  0 nni operations
## Final p-score 414 after  0 nni operations
## Final p-score 430 after  0 nni operations
## Final p-score 395 after  1 nni operations
## Final p-score 432 after  0 nni operations
## Final p-score 385 after  0 nni operations
## Final p-score 384 after  0 nni operations
## Final p-score 406 after  0 nni operations
## Final p-score 427 after  4 nni operations
## Final p-score 458 after  2 nni operations
## Final p-score 389 after  1 nni operations
## Final p-score 429 after  1 nni operations
## Final p-score 389 after  0 nni operations
## Final p-score 409 after  0 nni operations
## Final p-score 373 after  0 nni operations
## Final p-score 404 after  0 nni operations
## Final p-score 410 after  0 nni operations
## Final p-score 398 after  1 nni operations
## Final p-score 417 after  0 nni operations
## Final p-score 381 after  0 nni operations
## Final p-score 367 after  0 nni operations
## Final p-score 407 after  1 nni operations
## Final p-score 384 after  0 nni operations
## Final p-score 389 after  1 nni operations
## Final p-score 418 after  1 nni operations
## Final p-score 414 after  1 nni operations
## Final p-score 398 after  0 nni operations
## Final p-score 418 after  1 nni operations
## Final p-score 421 after  0 nni operations
## Final p-score 462 after  1 nni operations
## Final p-score 436 after  0 nni operations
```

```
## Final p-score 426 after  0 nni operations
## Final p-score 382 after  1 nni operations
## Final p-score 431 after  1 nni operations
## Final p-score 430 after  0 nni operations
## Final p-score 388 after  1 nni operations
## Final p-score 447 after  1 nni operations
## Final p-score 396 after  1 nni operations
## Final p-score 383 after  1 nni operations
## Final p-score 405 after  0 nni operations
## Final p-score 414 after  0 nni operations
## Final p-score 416 after  1 nni operations
## Final p-score 396 after  0 nni operations
## Final p-score 424 after  0 nni operations
## Final p-score 432 after  0 nni operations
## Final p-score 392 after  1 nni operations
## Final p-score 383 after  0 nni operations
## Final p-score 421 after  1 nni operations
## Final p-score 397 after  0 nni operations
## Final p-score 427 after  0 nni operations
## Final p-score 428 after  0 nni operations
```

For the bootstrap the columns/sites in the alignment get resampled.

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(opt_sim$tree, bootstrapped, type = 'phylogram', p = 40)
```



We can infer from the tree that nurse "Mme S" is more likely to have infected "Mme L" than "Mr D". In 97% of the 100 bootstrapped trees "Mme S" and "Mme L" are in the same cherry. "Mr D" is further away from "Mme L" in the tree and the support for the position for "Mr D" is only 48% of the bootstraps.