

Model Fitting and Ancestral Character State Reconstruction

Aim: Test if a simulated character evolves under symmetric or asymmetric Markov model of character evolution using AIC. Reconstruct its ancestral states.

```
library(ape)
```

Calculating Likelihood on tree

See lecture 6 for Felsenstein's pruning algorithm.

The general steps of the algorithm are as follows:

1. Initialize Q matrix and π (initial vector)
2. Recode data at tips as probabilities
3. Calculate conditional likelihoods L_c at each node by traversing the tree from tips to roots. For two branches – left and right (b_l and b_r) – and their corresponding tip probabilities (T_l , T_r , both column vectors), the conditional likelihood is $L_c = e^{Qb_l}T_l * e^{Qb_r}T_r$. Since R does not have inbuilt functionality for column vectors, T_l , T_r can be treated as row vectors through the following conversion $L_c = T_l(e^{Qb_l})^T * T_r(e^{Qb_r})^T$, where T is a matrix transpose
4. Upon reaching root, the tree likelihood is $L = \text{sum}(L_c\pi)$

```
# install packages expm if you do not have it
library (expm)
```

1. Initializing Q and pi

```
## Loading required package: Matrix
##
## Attaching package: 'expm'
## The following object is masked from 'package:Matrix':
##
##      expm
## The following object is masked from 'package:ape':
##
##      balance
Q <- matrix(
  c(
    -0.1, 0.1,
    0.1, -0.1
  ), 2,2, byrow = T)
pi=c(0.5, 0.5)
Q
##      [,1] [,2]
```

```
## [1,] -0.1  0.1
## [2,]  0.1 -0.1
```

```
pi
```

```
## [1] 0.5 0.5
```

```
# read in tree
```

```
tr <- read.tree(file='3_taxon.tree')
```

```
# Let's imagine that our taxa at tips (t1, t2, t3) have the following character states (we use a binary
```

```
data <- c(1,0,1)
```

```
names(data) <- c('t1', 't2', 't3')
```

```
data
```

2. Recoding data at tips as probabilities

```
## t1 t2 t3
```

```
##  1  0  1
```

We rate their tip probability vectors as matrix

```
data.probs <- matrix(c(
```

```
  0,1,
```

```
  1,0,
```

```
  0,1
```

```
), 3,2,byrow=T
```

```
)
```

```
rownames(data.probs) <- c('t1', 't2', 't3')
```

```
data.probs
```

```
##      [,1] [,2]
```

```
## t1      0      1
```

```
## t2      1      0
```

```
## t3      0      1
```

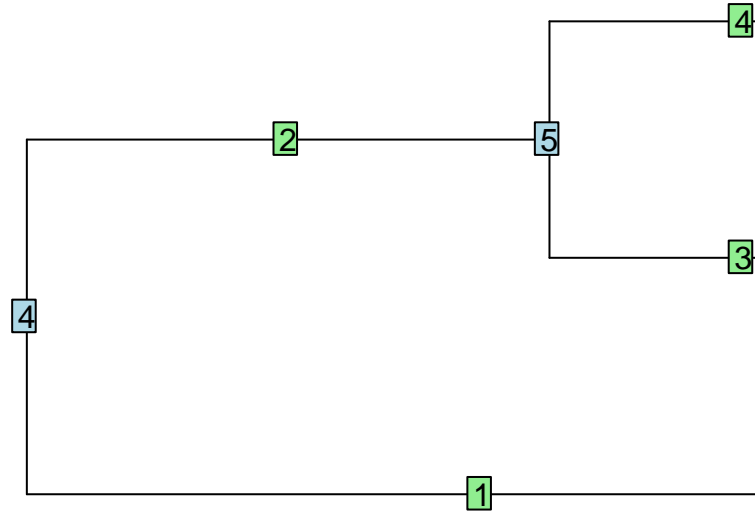
```
# Let's plot a tree to see its structure that is encoded on the objects tr$edge and tr$edge.length
```

```
plot(tr)
```

```
nodeLabels()
```

```
edgeLabels()
```

```
tiplabels()
```



3. Calculating conditional likelihoods

First we calculate Lc for node 5 with tips t3 and t2

```
# get ids of two branches node5 -> t3 and node5 -> t2 in tr$edge object (they are 3 and 2)
t3t2.ids=which(tr$edge[,1]==5)

# get branch lengths for those branches
br_l_r=tr$edge.length[t3t2.ids]

# calculate conditional likelihood using matrix exponential expm(), and matrix multiplication %*%
L5=data.probs[3,] %*% t(expm(Q*br_l_r[1]))*(data.probs[2,]%*%t(expm(Q*br_l_r[2])))
L5
```

```
##           [,1]           [,2]
## [1,] 0.03916016 0.03916016
```

Now we calculate root likelihood (node4)

```
# get ids of two branches node4 -> t1 and node4 -> node5
n5t1.ids=which(tr$edge[,1]==4)
# get branch lengths for those branches
br_l_r=tr$edge.length[n5t1.ids]
br_l_r
```

```
## [1] 1.0000000 0.5740947
```

```
# calculate conditional likelihood at root using matrix exponential expm(), and matrix multiplication %*%
Lr=(data.probs[1,]%*%t(expm(Q*br_l_r[1])))*(L5 %*% t(expm(Q*br_l_r[2])))
Lr
```

```
##           [,1]      [,2]  
## [1,] 0.003549267 0.0356109
```

```
L=sum(Lr*pi)
```

```
# likelihood and Log likelihood of the tree  
L
```

4. Calculating tree likelihood

```
## [1] 0.01958008
```

```
log(L)
```

```
## [1] -3.933242
```