2.4 Inferring Phylogeny using Maximum Likelihood in R (phangorn)

Background

Maximum likelihood (ML) is based upon calculating the probability of observed data given a hypothesis. The alternative hypotheses in phylogenetic inference are all the various trees that can be drawn for a set of taxa. In an ML search, we aim to find the tree that, given our evolutionary model, results in the highest likelihood of obtaining the data we observe. Thus, we are maximizing the likelihood of the data under the model for evolution that we choose, which is covered in more detail in Section 2.5.

ML is a different approach to tree inference than parsimony, but there are important similarities. For example, both ML and parsimony search tree space in similar ways. When trees become large, one must often devise a strategy for sampling a subset of all possible trees, because no computer is fast enough to sample the huge number of possible trees, which increase exponentially as a function of number of taxa. In addition, ML shares with parsimony the goal of finding the "best" tree topology. The difference is that, rather than finding the tree that minimizes the number of inferred changes, ML finds the tree that maximizes the likelihood of the data.

That sounds confusing at first, but can be understood if you consider that simply counting changes fails to make use of all the information available. For example, the parsimony method effectively assumes that the probability of a change happening is the same regardless of how long ago two taxa diverged. This, of course, is an unrealistic assumption, because taxa that diverged in the more distant past have had more time for evolutionary changes to take place, assuming there is a roughly constant probability of change over time. ML uses an explicit probability model that can effectively account for the greater number of evolutionary changes that are expected on long branches relative to short ones (see Section 2.5).

Maximum likelihood example in R

In later sections, we will use R and other programs to select a model of evolution, and as part of that process, we will infer a phylogeny using maximum likelihood. Before proceeding, however, it is worth noting that the R package 'phangorn', which was used in the previous two sections, provides some simple tools to compare the likelihood of the data under different models of evolution or among different phylogenies. Keep in mind that we are using a small dataset of just 150 nucleotides, and this is not meant to be a definitive analysis. Rather, this example simply illustrates that many fundamental steps of a maximum likelihood analysis can be conducted easily in R. First, load the two packages we will need for this section (see subsection 1.1.3 for instructions on installing packages):

```
library("phangorn")
```

The phangorn function 'pml' provides a way to compute the likelihood of the data given a phylogenetic tree and evolutionary model. Drawing on the trees obtained in Section 2.2, let's do some simple likelihood computations (you will need to first complete Section 2.2 to run these analyses).

```
fit_treeUPGMA = pml(unroot(treeUPGMA), data=primates)
```

Type 'fit_treeUPGMA' to obtain the key statistics, such as the likelihood and the transition matrix that is assumed, and 'summary(fit_treeUPGMA)' to see the output format of type 'pml'. We can then optimize the branch lengths under this simple model of evolution (the Jukes-Cantor model, where all changes are equally likely), and compare the two trees graphically:

```
fit_treeUPGMA_opt1 = optim.pml(fit_treeUPGMA)
layout(matrix(c(1,2)), height=c(1,1))
par(mar = c(.1,.1,.1,.1))
plot(fit_treeUPGMA, main="default branches", cex = 0.8)   # top = default branch lengths
plot(fit_treeUPGMA_opt1, main="optimized branches", cex = 0.8)   # bottom = optimized branch
lengths
```

In the graphics window, the tree on top shows the default branches, while the tree on the bottom shows the branches after optimization. Longer branches indicate that more molecular change has occurred along a given branch. You will find that the likelihood of the data under optimized branch lengths has increased considerably - from 1573.4 to 1555. We can compare these models using the 'AIC' function and we find unsurprisingly that the AIC is substantially lower for the tree with branches optimized.

```
AIC(fit_treeUPGMA, fit_treeUPGMA_opt1)
```

We can also search for a better tree by re-arranging the branches, i.e., by setting the parameter 'optNni' to 'TRUE', which causes the function 'optim.pml' to optimize tree topology in addition to branch lengths:

```
fit_treeUPGMA_opt2 = optim.pml(fit_treeUPGMA, optNni=TRUE)

layout(matrix(c(1,2)), height=c(1,1))

plot(fit_treeUPGMA_opt1, cex = 0.8)  # top = original topology with optimized branch lengths

plot(fit_treeUPGMA_opt2, cex = 0.8)    # bottom = optimized topology AND branch lengths

AIC(fit_treeUPGMA_opt1, fit_treeUPGMA_opt2)
```

This slightly revised tree drops the AIC almost 10 points, indicating better fit. Many other functions can be used to further optimize parameters of the evolutionary model, the tree, or to test alternative models of evolution (e.g. a generalized time reversible model). In the following section, we further explore model selection using R in combination with another program called PHYLM.

References

> *Bolker, B., and R Development Core Team (2011). bbmle: Tools for general maximum likelihood estimation. R package version 0.9.7.* http://CRAN.R-project.org/package=bbmle

> *Felsenstein J. 2004.* Inferring Phylogenies*. Sunderland MA: Sinauer Associates.*

> *Hodgson, J. A., K. N. Sterner, L. J. Matthews, A. S. Burrell, R. L. Raaum, C. B. Stewart, and T. R. Disotell. 2009. Successive radiations, not stasis, in the South American primate fauna.* Proceedings of the National Academy of Sciences, USA*. 106:5534-5539.*

> *R Development Core Team (2010). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL* http://www.R-project.org*.*

> *Schliep, K. 2010. phangorn: Phylogenetic analysis in R. R package version 1.2-0.*

Contributed by Luke Matthews, Charlie Nunn and Randi Griffin