Lecture 13 – Consensus Trees & Nodal Support

I. Introduction: There are a number of methods that can be used to assess how strongly a data set supports a particular relationship.

These make use of consensus trees, so we need to spend a little time describing them

II. Consensus Trees – Chapter 30 in Felsenstein’s book

Any time you have more than a single data set, you may wish to compare the trees that analyses of the separate data produce. This has long been done using consensus trees.

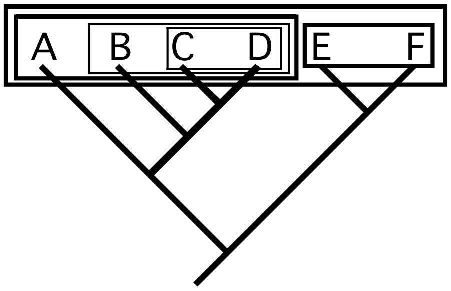
These are **best treated as visual summaries** of the agreement and disagreement between (among) source trees, and consensus trees can be generated from any number of trees (> 1).

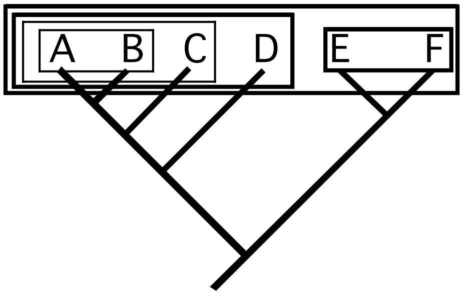
These source trees may come from analyses of multiple data sets, they may be trees produced by analyzing the same data set with different methods, or they may be equally optimal trees (i.e., there may be many trees of the same length or with the same likelihood score). Critically, they’re used to summarize distributions of trees (generated via MCMC or bootstrap).

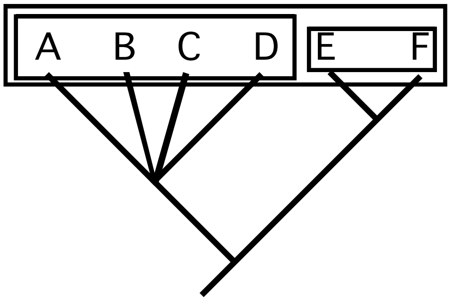
There are several types of consensus trees:

A. Strict Consensus Trees.

A strict consensus tree contains only groups that are *exactly* represented in all the input trees. Thus, this is the most conservative consensus method. Computing these by hand is simple to do by using Venn Diagrams.







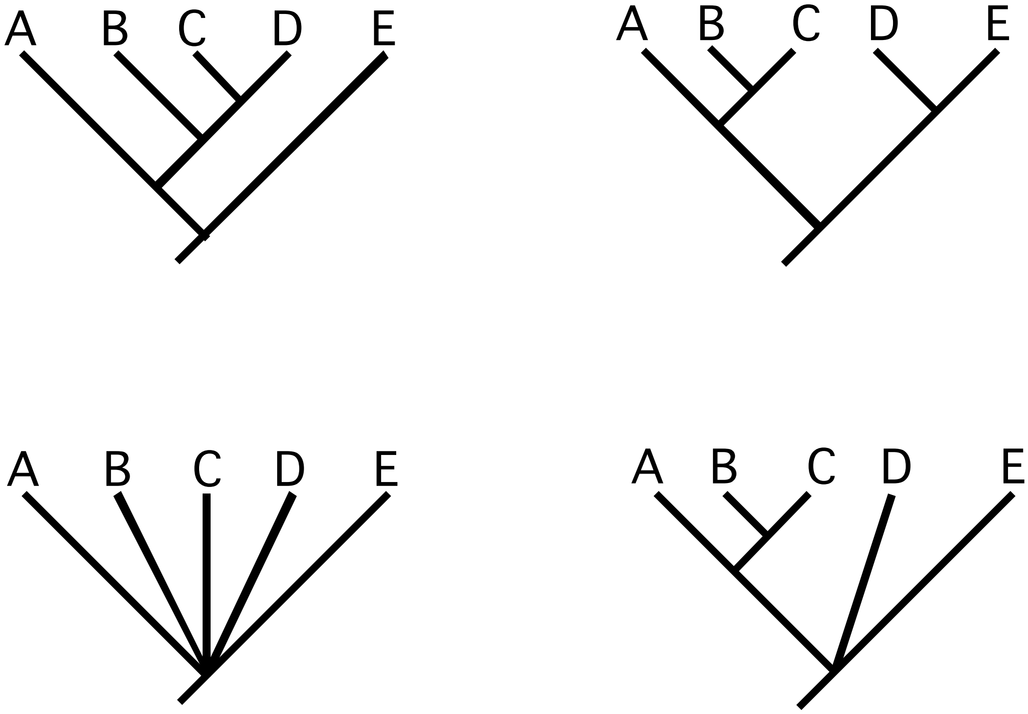
The ellipses of the Venn diagrams represent the groupings on the trees we're comparing.

Only the **ellipses in bold are present on both the trees**, so only the groupings that those ellipses encompass are present in the strict consensus tree; strict consensus trees are the least resolved consensus trees.

B. Adams Consensus Trees

At the other end of the spectrum are the Adams consensus trees, which are designed to maximize resolution in the consensus.

They essentially work by **congruence of three-taxon statements** and simply relocate offending taxa. This is why they have more resolution than strict consensus trees.



Strict ConTree

Adams ConTree

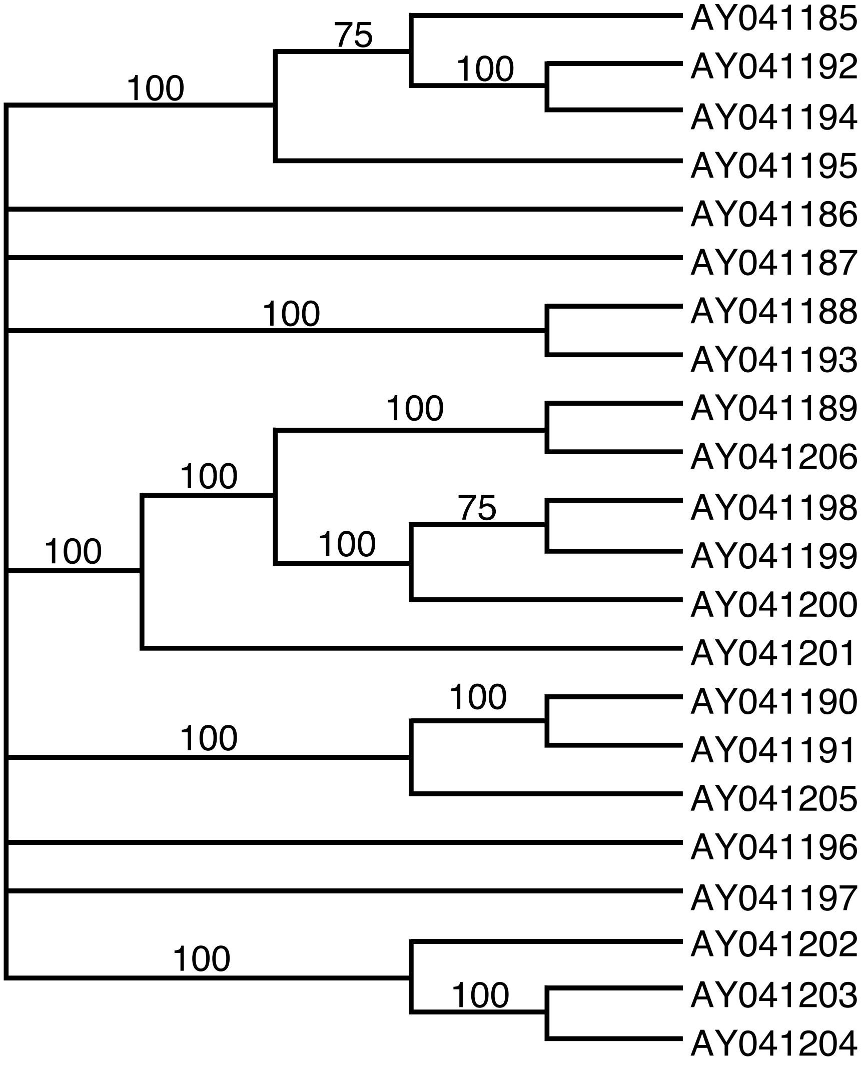
So, the strict consensus of these two trees is completely unresolved, whereas the Adams consensus has two nodes. **Both these nodes unite groups {A, B, C} and {B, C} that are not present in both the initial trees.**

Adams consensus trees can only be used for rooted trees.

C. Majority Rule Consensus Trees

These are exactly what their name implies. Nodes that occur in the majority of the trees being compared are left resolved in the majority rule consensus tree.

Below is the majority rule consensus tree of the NJTree (LogDet distances), MP trees (2), and the ML tree for the data set from lab. So, all four trees have the nodes indicated by the 100's, etc.



These are pretty meaningless as nodal support values if they're derived from a collection of MP trees or trees produced by different methods.

Again, we end up accepting groups that do not occur on all the trees being compared.

However, majority rule consensus tree do serve a critical function in estimating nodal support using two common measures.

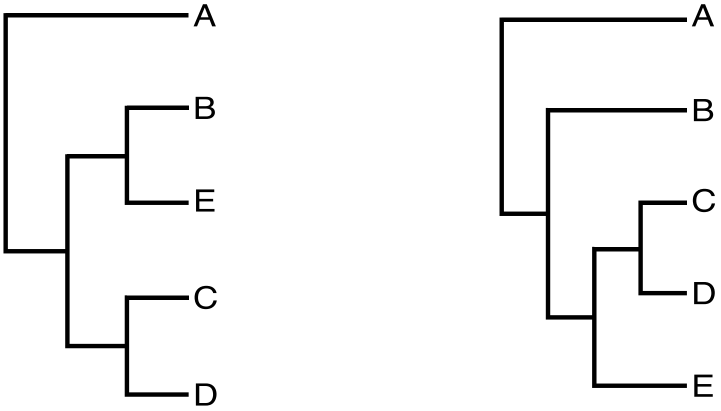
D. Before we discuss how majority rule consensus tree are used in estimating nodal support, I want to make a **remark regarding consensus trees**.

These do not represent phylogenies. As such, they should never be shown with branch lengths, and characters should not be optimized on them.

To see why, let's use a parsimony example and look at this data matrix:

A	0	0	0	1	0	0
B	1	0	1	0	0	0
C	0	1	0	1	1	1
D	0	1	0	1	1	1
E	1	1	1	1	1	0

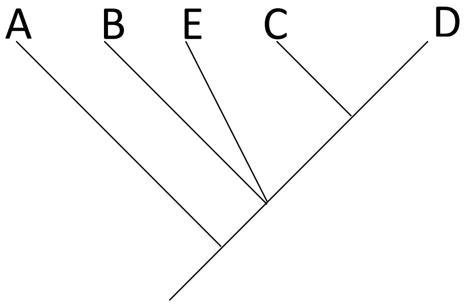
There are two MP trees for this matrix:



These are each 8 steps.

Characters 2 & 5 are homoplasious.

Characters 1 & 3 are homoplasious.



If we do the wrong thing, by treating the consensus tree as a phylogeny and map the characters onto it, all four of those characters are homoplasious and the ConTree is 10 steps. **This extends to likelihood, as well**.

Therefore, one should never show branch lengths on a consensus tree. You should demonstrate this to yourselves and I'll put this in the problem set (with its answer).

III. Nodal Support – Most often, the question of greatest interest has to do with how well supported are the various groups that are present in the optimal topology.

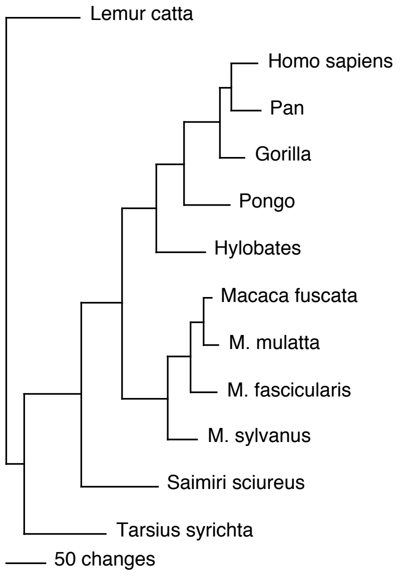
Again, methods that have been developed to assess this are explicitly statistical.

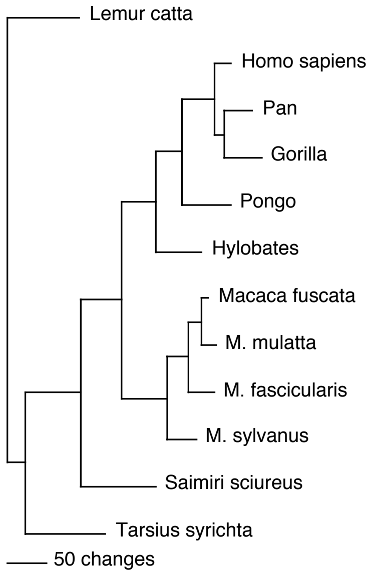
A. Decay Index (a.k.a., Bremer Support)

Bremer (1988. Evolution, 42:795) developed a parsimony approach to assess nodal support from a non-statistical perspective. The idea is that if an MP tree is, say 996 steps, and group (A, B, C) is found on that tree, we may wish to know how much longer is the best tree that doesn't contain group (A, B, C). This is the decay index for that group.

So, for the primate mtDNA data set (Hayasaka et al., 1988. Mol. Biol. Evol., 5:626) that is often used as a sample data set there are two MP trees, each of length 996 steps.

On one tree, *Homo* & *Pan* are sister taxa whereas on the other *Pan* and *Gorilla* are sisters.

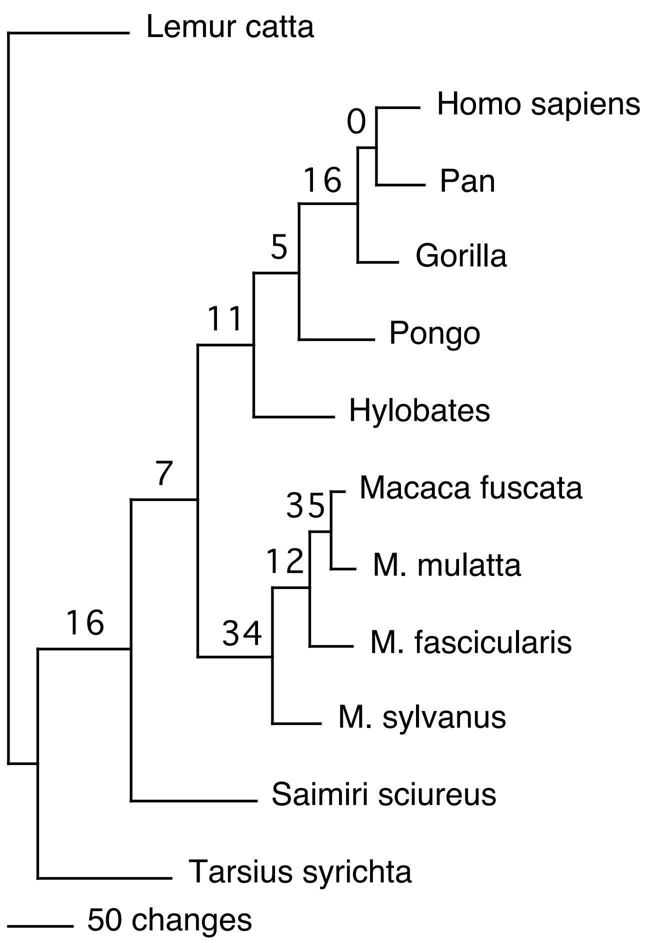




Therefore, the each of these two groups has a decay index of 0.

The group (*Homo*, *Pan*, *Gorilla*) occurs on both. The shortest tree that doesn't contain this group is 1012 steps. Therefore, the *Homo*/*Pan*/*Gorilla* node has a decay index of 16.

There are a couple ways to do this, and it can be done for each node, to give the following:



Each node has a decay index associated with it that indicates how much longer the shortest tree is that doesn't contain that particular node.

This certainly is an advantage over the other methods that we've been discussing in that we now have an assessment of how strongly our data supports each particular hypothesis of relationships. Those that have higher decay indices are more strongly supported.

However, these are just numbers, and it's very difficult to decide how large a decay index is meaningful. Perhaps the best way to think about this is that the decay index for a node indicates how many new, conflicting synapomorphies would need to be discovered to overturn this hypothesis of monophyly.

B. Bootstrap Support.

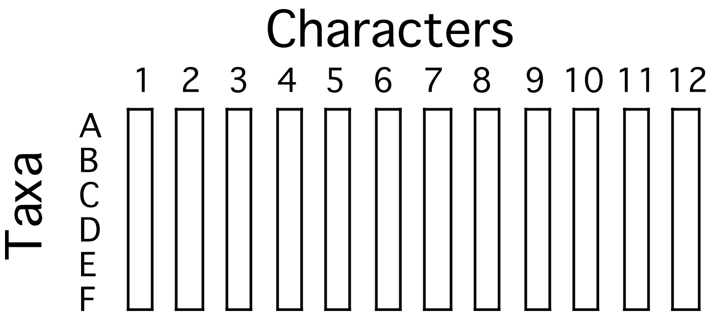
Felsenstein (1985. Evolution, 39:783) was the first to suggest using the bootstrap approach to assessing nodal support in phylogenies.

For those unfamiliar with it, there's an excellent introduction to it from a general perspective on pages 345 & 346 in the text.

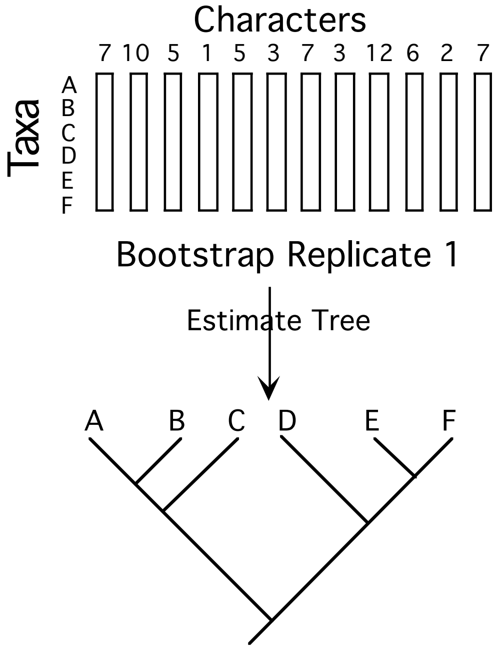
The method was developed by Efron in 1979 to develop confidence intervals in cases where the true underlying distribution of a variable can't be assessed. The idea is that the distribution of the original sample (*if it's large enough*) will convey much about the nature of the underlying true distribution.

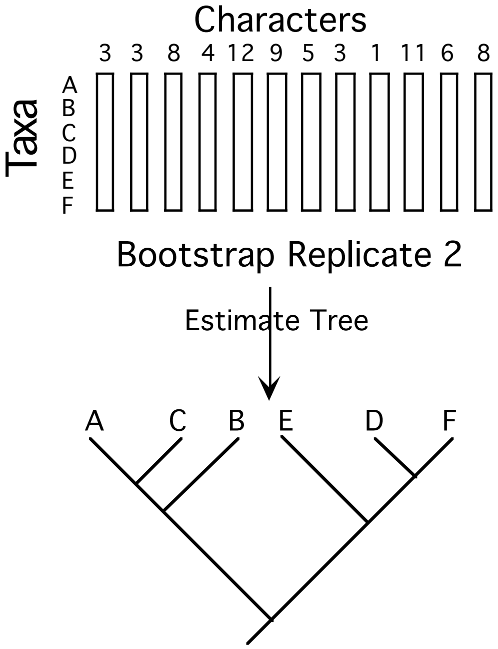
So, we can treat our original sample as if it were the true distribution (it certainly estimates the true distribution) and take repeated samples of our original sample to mimic the variability we would see if we could resample from the true distribution.

In the case of phylogenies, we are interested in resampling characters from the original sample (i.e., the data) that we have collected. We then treat that sample of characters as estimating some underlying true distribution of characters.



The columns (characters) are re-sampled with replacement, and usually each pseudo-replicate is the same size as the original data set.





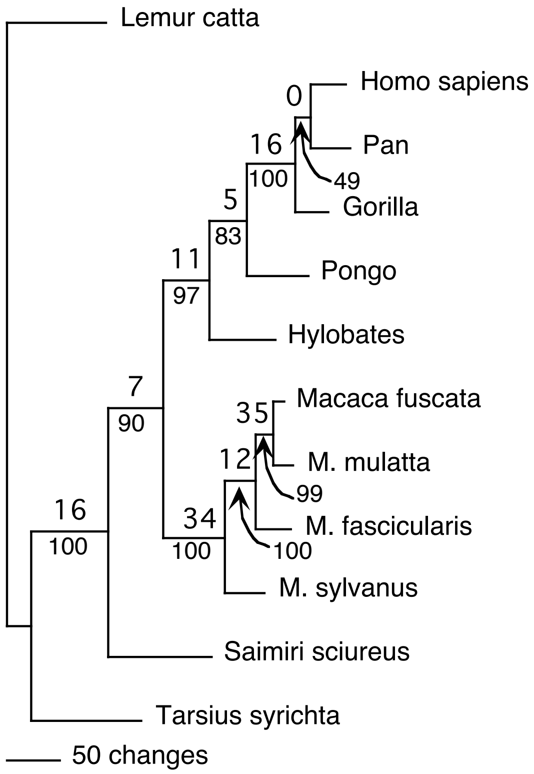
We then generate some large number (usually between 100 & 2000) of pseudo-replicate data sets by randomly selecting characters with replacement from the original sample.

We estimate the phylogeny for each of these pseudo-replicate data sets to generate a collection of trees. These can be generated under any of the optimality criteria we've discussed, and the variation among these trees provides a measure of uncertainty in our original phylogenetic estimate.

A majority-rule consensus tree is used to show the percentage of bootstrap replicates in which any particular node is found in the ML, MP or ME tree for that replicate.

This percentage is the bootstrap value for that node.

This is one of the two MP trees for the primate data, with MP bootstrap values indicated below each branch, and decay indices above the branch. There's a pretty good correlation between DI and bootstrap values.



There are several things we need to point out:

1) The size of the original sample is a critical factor in the performance of the bootstrap. This makes intuitive sense. The original sample is taken as a proxy for the underlying parametric distribution; it must be large enough to reflect relevant features of the distribution.

2) For large data sets, bootstraps can take a very long time. Felsenstein (1993 Phylip Manual) has suggested that the uncertainty captured by bootstrap resampling is much larger than the error associated with extensive branch swapping in estimating optimum tree for each bootstrap replicate.

At one extreme, one may conduct a full heuristic search for each bootstrap replicate (including stepwise addition with multiple random addition sequences and TBR branch swapping).

An intermediate strategy that has been shown to work well for parsimony (DeBry & Olmstead, 2000. Syst. Biol., 49:171) is doing a greedy search on each bootstrap replicate that involves retaining only a single optimal tree in memory (i.e., MAXTREES = 1). This was explored for ML analyses empirically by Ripplinger et al. (2010. MP&E. 56:642) who found the same result (i.e., no need to worry about model selection/parameter optimization for each pseudoreplicate; cursory branch swapping should be done).

One could do a FastBoot analysis, in which only a stepwise-addition tree is built for each bootstrap replicate. This is analogous to a neighbor-joining bootstrap.

IQ-TREE does UltraFastBoot. This uses the RELL bootstrap and doesn't do any tree searching on bootstrap replicates. I'll teach about RELL later and we'll address this then.

3) As long as we're using a consistent estimator of phylogeny, bootstrap values on nodes tend to be conservative as confidence intervals (Hillis & Bull, 1993. Syst. Biol., 42:182). If we're using an inconsistent estimator, of course bootstrap analysis may give us high confidence in incorrect nodes (i.e., if we're in the Felsenstein Zone), or they may give us too much confidence in a correct node that is actually poorly supported (i.e., if the true tree is in the inverse Felsenstein zone).

Lots of work has been done on how to interpret the bootstrap values, and Joe does an excellent job summarizing that work on pages 335 – 345.

My take is that they can be taken as estimates of the statistical confidence we can place in a node (or anything else we may be estimating using the phylogeny), but that in some cases they're conservative and in other cases they're too liberal. As such, we need to treat them cautiously, and think about conditions that lead to each type of bias.

Next, we'll discuss an alternative to the bootstrap in estimating nodal support, Bayesian Posterior Probabilities and we'll use that to introduce Bayesian statistics.

C. Bayesian Estimation of Nodal Support

Just in the last twenty or so years, Bayesian statistics have become mainstream in phylogenetics. I'll just give a brief introduction to Bayesian statistics for those of you who are not familiar with the approach.

The idea of Bayesian analysis is intuitively very appealing. Given some data, a likelihood model, a quantification of prior our knowledge, we can calculate the probability of that some hypothesis is true.

The formalization of this is provided by Bayes' Theorem:

$$P(H_i \mid D) = \frac{P(H_i)P(D \mid H_i)}{\sum_{i=1}^{s} P(H_i)P(D \mid H_i)},$$

where $P\,(H_i \mid D)$ is the posterior probability of hypothesis i, given the data, D.

$P\,(H_i)$ is the prior probability of hypothesis i, (this is the quantification of prior knowledge).

$P\,(D \mid H_i)$ is the probability of the data, given hypothesis i. This is the regular likelihood function we've been using this semester.

The denominator is the product of these summed across all s competing hypotheses.

For phylogeny estimation, we can describe Bayes' Theorem as:

$$P(\tau_i \mid D) = \frac{P(\tau_i)P(D \mid \tau_i)}{\sum_{i=1}^{s} P(\tau_i)P(D \mid \tau_i)}$$

The $P\,(\tau_i)$, the prior probability of tree i, is usually set to $1/s$, where s is the number of possible trees. This represents an admission of ignorance and is called a flat prior or a uniform prior (or an uninformative prior).

The summation in the denominator then is across all s topologies.

Before we move on any further, I want to insert a comment as to why this is such an intuitively appealing approach.

First, it is literally the issue of actual interest for which we're calculating posterior probabilities: the (conditional) probability that a hypothesis is correct. This is in contrast to the frequentist probability of observing the data, given the hypothesis is true.

Second, this really captures the process that our minds go through when we read a scientific paper. We come to the paper with some background knowledge about the topic, which is analogous to the prior.

Maybe we have a great deal of expertise in the issue, in which case we would have an (informal) informative prior that may have a large impact on our belief after reading the paper (i.e., is the paper b.s., considering the data and our background knowledge?).

Conversely, we may have very little expertise on the topic, in which case our (informal) prior would not be very informative (i.e., flat) and have very little impact on our belief after reading the paper. So, every time we read a scientific paper, our minds conduct an informal Bayesian analysis.

Let's go back to Bayes' Theorem as it applies to phylogenies. The denominator is impossible to compute. In order to calculate it, we need to calculate the likelihood of all possible trees. This rendered fully Bayesian analysis impossible for the 25 years between the advent of computational phylogenetics and the paper by Ranala and Yang (1996. J. Mol. Evol., 43:304) and Mau's dissertation (1996) that led to the current ascendency of Bayesian estimation.

This is due to the application of Markov Chain Monte Carlo to Bayesian estimation.

MCMC is an approach that allows one to derive a sample from an unknown distribution by growing a chain of states that are sampled from this distribution. This is accomplished by proposing a change to the current state and accepting that proposal following a set of rules.

For phylogenies, we start the mcmc with some tree (τ_i), let's say it's a random tree. This is the state of the chain in the first generation.

We then propose a change in the tree (generate τ_{i+1}) by proposing a random NNI (or some other type of tree rearrangement - see below).

If the new tree has a higher posterior probability than the first, we accept the new tree.

This decision is made by calculating the ratio of the posterior probability of the new to previous state (tree):

$$R = \frac{\dfrac{P(\tau_{i+1})P(D \mid \tau_{i+1})}{\sum_{i=1}^{s} P(\tau_i)P(D \mid \tau_i)}}{\dfrac{P(\tau_i)P(D \mid \tau_i)}{\sum_{i=1}^{s} P(\tau_i)P(D \mid \tau_i)}}$$

So, if R > 1, we accept the new tree (it has a higher posterior probability than the previous tree).

If R < 1, we draw a random probability (between 0 & 1). If this is > R, we accept the change, if not, we return to the previous state (tree). We'll accept slightly worse trees more often than we'll accept much worse trees. This builds down-slope movement into MCMC.

By examining the acceptance ratio, we can see a couple of simple things.

First, the impossible denominators for each state cancel; we never have to compute the impossible denominator.

Second, if the priors are the same across all topologies, that is,

$$P\ (\tau_{i+1}) = \mathrm{P}\ (\tau_i) = 1/s,$$

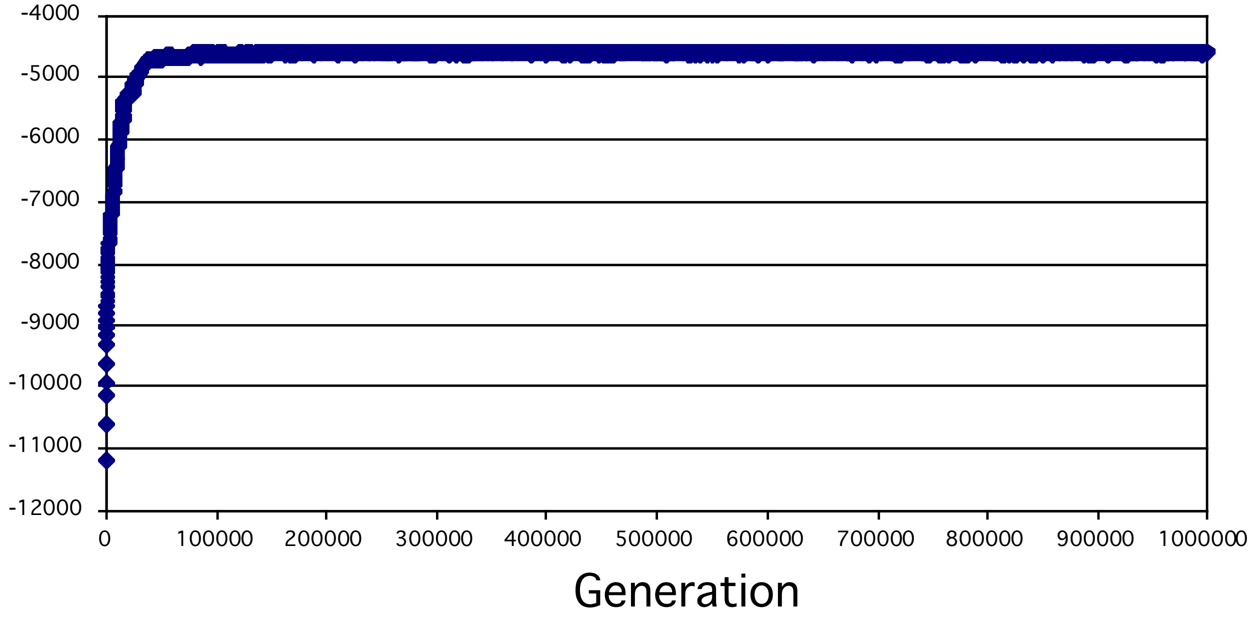
The priors cancel and the likelihood function determines the shape of the posterior probability distribution.

Using these rules, and starting anywhere in tree space, if we run the chain long enough, eventually the frequencies of states in the chain (i.e., trees), will converge to their frequencies in the posterior probability distribution.

Another way of saying this is that once the chain reaches equilibrium, it samples trees proportionally to their posterior probability, and the sample provides a representation of the posterior distribution.

Now we need to make a few points.

First, we have to discard early generations of the chain, because it takes a while for the chain to traverse solution space and converge to the target distribution. This is called the burn-in, and a first-approximation of the burn-in usually entails a plot of the likelihood of the current tree across generations.



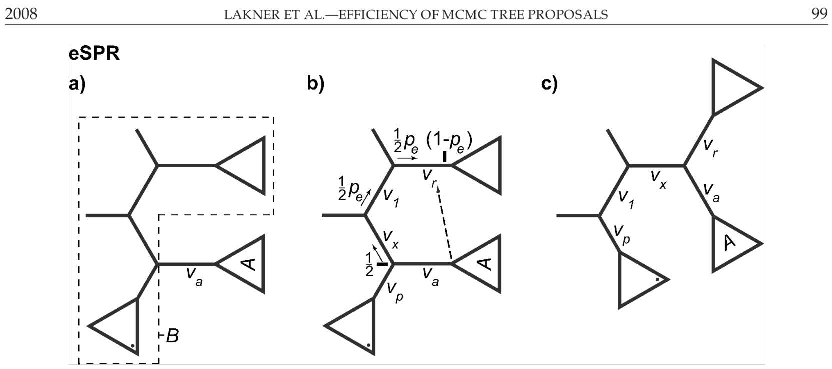
Here, once we've run the chain for ca. 100K generations, we seem to have converged, at least with respect to the likelihood.

Second, all that matters in theory is that all the potential manifestations of the states (i.e., all trees) can be reached by a series of single steps of our proposal mechanism (e.g., SPR branch swaps). As long as this is the case, the MCMC will converge on the stationary probability distribution if it's run long enough. *That is, MCMC will be ergodic*.

Nevertheless, convergence properties depend on proposal mechanisms. Proposal smechanisms that change the state very little will take very (very, very….) long chains to provide an adequate sample because solution space will be explored too narrowly. Conversely, proposal mechanisms that change the state too dramatically will result in most proposals being rejected and the chain sticking on a state for a long time.

Too little has been published on the effect of proposal mechanisms on Bayesian estimation of phylogenies, but Lakner et al. (2008. Syst. Biol. 57:86) is a great start.

They evaluated seven types of topology proposals for several empirical data sets. These included local branch changes that collapse branch lengths, and variations on NNI, SPR, and TBR swaps.



The restrictions focus reattachment points close to the original location. Internal branch $\mathbf{v_a}$ is broken and subtree A is attached on either side of its initial attachment point with probability of 0.5. It will be attached farther along with probability 0.5$\mathbf{p_e}$, where $\mathbf{p_e}$ is the extension probability.

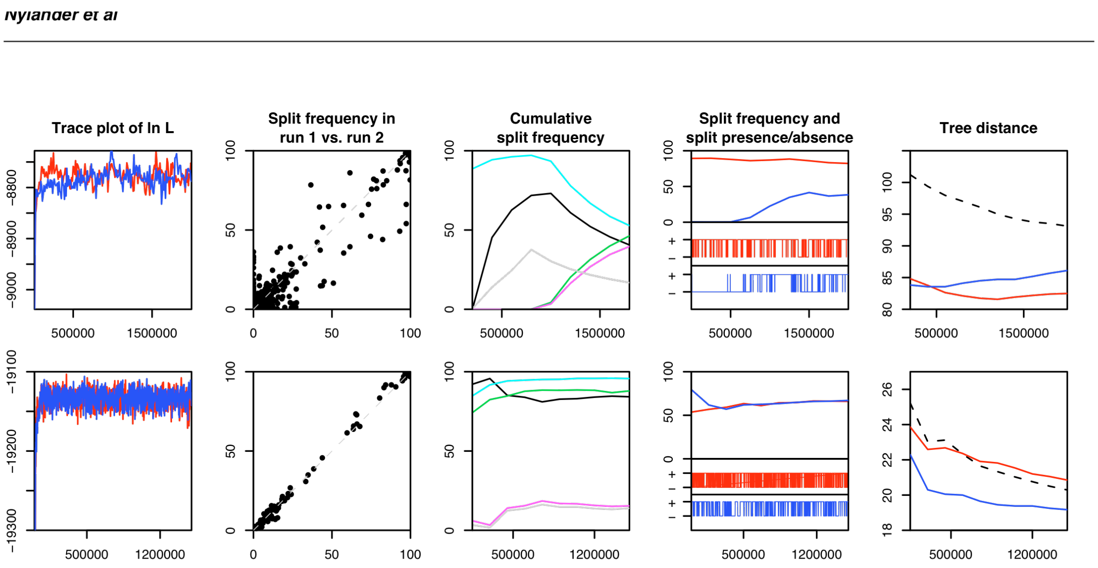
eTBR works the same way except the subtree A can be rotated to reconnect it at any point within it, with a higher probability of reconnecting the subtree at branches close to $\mathbf{v_a}$.

These restricted versions of branch swapping don't suffer from proposing changes that are so different that they're never accepted (which tends to happen with the non-restricted versions), nor do they suffer from proposing changes that are so small that solution space is traversed too slowly.

Proposal mechanisms that combine eTBR or eSPR with the local, small-scale changes seem to converge most rapidly.

Third, convergence diagnostics must address the parameter of interest. For example, in the plot above, the likelihood seems to have converged, but that tells us nothing about convergence with respect to topology.

Nylander et al. (2008. Bioinformatics, 24:581) have produced Are We There Yet (AWTY), to focus convergence diagnostics on a number of potential parameters of interest.



These pretty sophisticated convergence diagnostics do not demonstrate convergence, but are at least consistent with the hypothesis that independent runs are sampling from the same distribution, and we hope that this is the actual target posterior distribution of (in this case) topologies.

So, we can sample the tree every so often from this chain to generate a collection of trees (say 10,000 of them). Ideally, each topology will be present in the sample at a frequency that's proportional to its posterior probability.

Summarizing the posterior distribution of trees.

This generates a sample from the posterior distribution of trees, and the frequency of each tree in the sample is that tree's posterior probability.

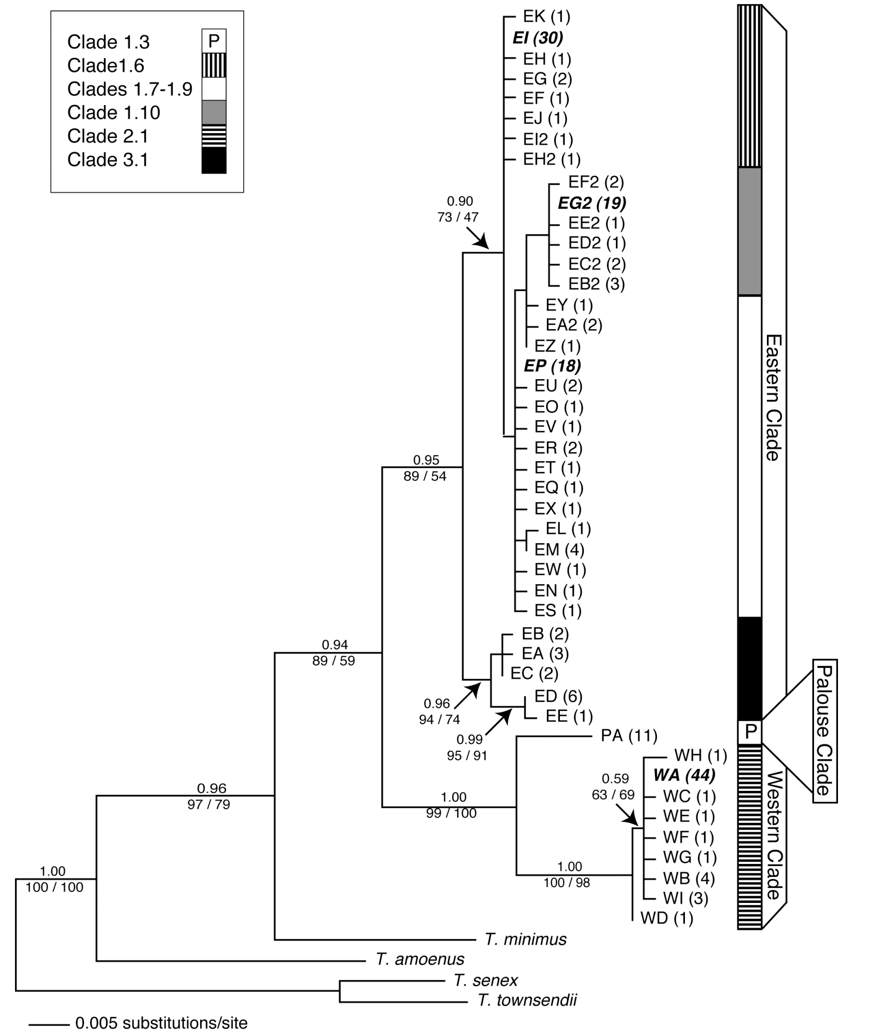
There are several ways to summarize the posterior distribution of trees.

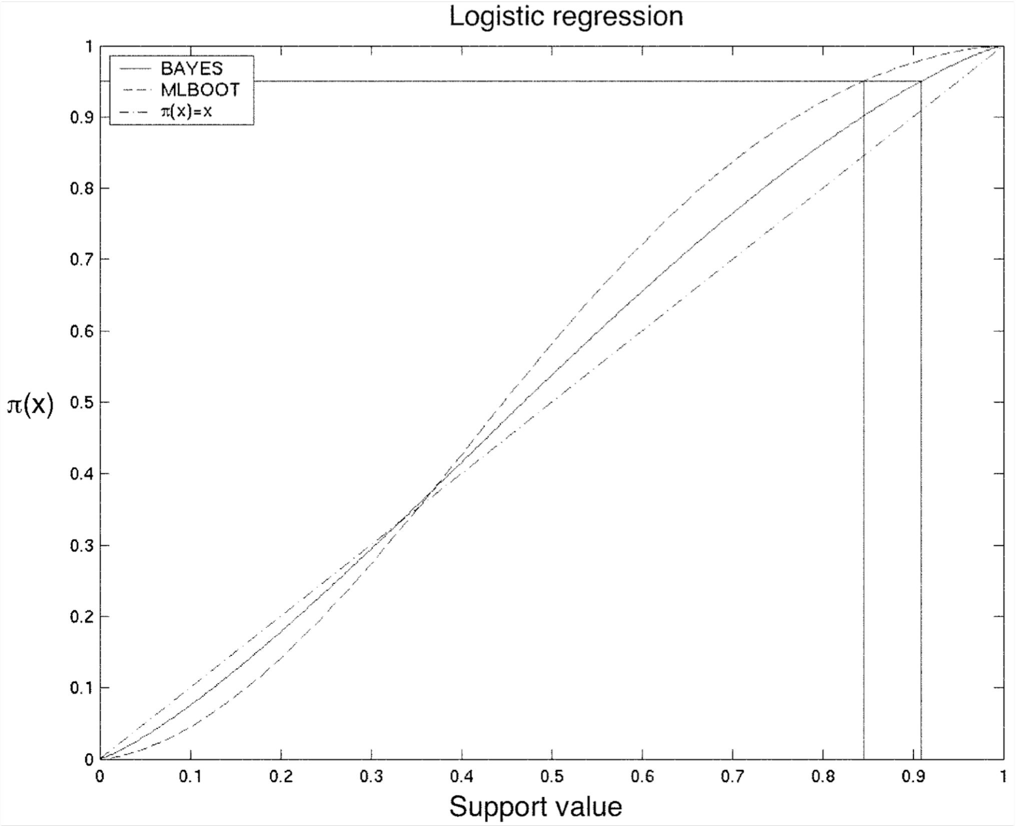
We can compute the majority-rule consensus tree from this distribution of trees to see how frequent each node (say on the ML tree) is in the sample. This becomes our posterior probability that the node is correct, conditional on the data, the priors, and likelihood model.

An example of this is shown below. Good & Sullivan (2001. Mol. Ecol., 10:2683):

So, as is commonly seen, there are nodes for which all three methods indicate strong support, and there are nodes for which the Bayesian posterior nodal probabilities are much higher than either the MP or ML bootstraps.

So, we can assess if the higher nodal support values are actually meaningful, and a fair amount of work has been done on it. For example, Erixon et al. (2003. Syst Biol. 52:665) suggest (as do other studies - Alfaro et al. 2003. Mol. Biol. Evol. 20:255) that under best-case scenario (correct model), both ML bootstrap values and BPP are conservative, but BPP are less so.





There are other ways to summarize the posterior distribution of trees, such as the MAP (Maximum posteriori) tree.

I actually think a far better use of the posterior distribution of trees is to use them to test hypotheses (more later) or sample from the posterior for a second-order analysis.

There are lots of **other issues** to deal with regarding Bayesian estimation.

First, the phylogeny problem is a terribly complex one. Remember that the likelihood of a particular tree topology includes a vector of branch lengths, each of which is estimated with uncertainty and a vector of model parameters, each of which is estimated with uncertainty.

So, we can take our model, start the mcmc with initial states for model parameters, and include proposals to change each of those as well as topology in our mcmc.

This means that we need to place priors on each of these parameters as well. Typically, MrBayes does something like this:

```
Parameters
      ------------------
      Revmat           1
      Statefreq        2
      Shape            3
      Pinvar           4
      Topology         5
      Brlens           6
      ------------------

      1 --  Parameter  = Revmat
            Prior      = Dirichlet(1.00,1.00,1.00,1.00,1.00,1.00)
      2 --  Parameter  = Statefreq
            Prior      = Dirichlet
      3 --  Parameter  = Shape
            Prior      = Uniform(0.05,50.00)
      4 --  Parameter  = Pinvar
            Prior      = Uniform(0.00,1.00)
      5 --  Parameter  = Topology
            Prior      = All topologies equally probable a priori
      6 --  Parameter  = Brlens
            Prior      = Branch lengths are unconstrained:
                         Exponential(10.0)
```

Proposed changes to these parameters are usually as follows:

```
The chain will use the following moves:
      With prob.  Chain will change
        3.57 %   param. 1 (revmat) with multiplier
        3.57 %   param. 2 (state frequencies) with Dirichlet proposal
        3.57 %   param. 3 (gamma shape) with multiplier
        3.57 %   param. 4 (prop. invariants) with beta proposal
       53.57 %   param. 5 (topology and branch lengths) with LOCAL
       10.71 %   param. 5 (topology and branch lengths) with extending TBR
       10.71 %   param. 6 (branch lengths) with multiplier
       10.71 %   param. 6 (branch lengths) with nodeslider
```

The advantage of this is that we’re estimating nodal probabilities that actually account for uncertainty in all the other parameters (i.e., integrate across uncertainty in estimating them).

In contrast in ML analysis, we’re using point estimates (the ML values) of each parameter (jointly estimated) and use of marginal probabilities incorporate this uncertainty intelligently. Holder and Lewis (2003. Nature Rev. Gen. 4:275) provide a great discussion of joint versus marginal estimation:

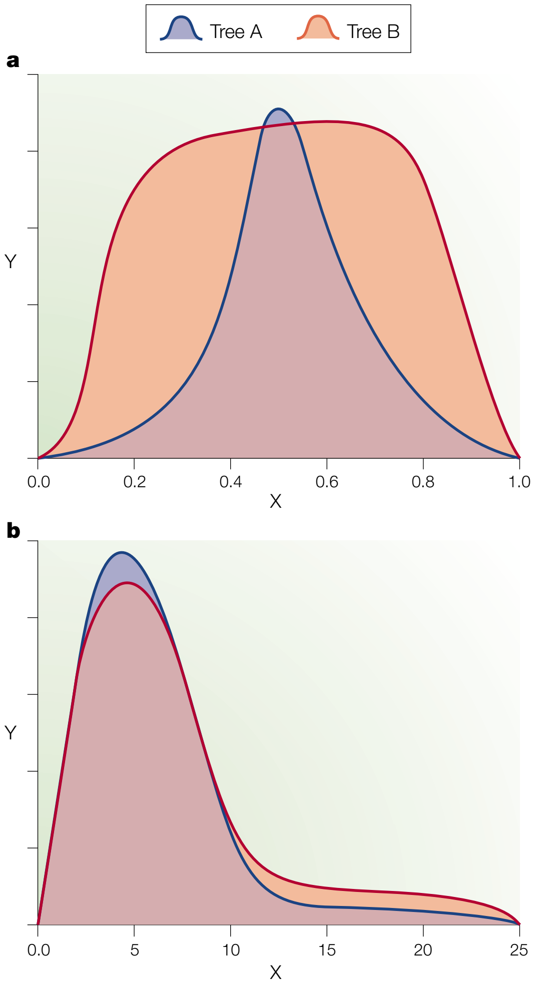


Figure 1 | **Contrast between marginal and joint estimation.** Panels **a** and **b** depict the likelihood profile for two trees versus a hypothetical parameter *x*. The *x* axis represents some nuisance parameter (for example, the ratio of the rate of transitions to the rate of transversions). The *y* axis represents the likelihood in the case of ML, or the posterior-probability density in a Bayesian approach. The area under the likelihood curve for tree A is shown in light blue, the area for tree B is shown in orange. Mauve regions are under the curve for both trees. In both cases, jointly estimating *x* and the tree favours tree A (that is, the highest peak is blue in both cases), but marginalizing over *x* favours tree B (that is, the orange area is greater than the blue area).

The disadvantage is that now we're generating an even more complex parameter space through which to traverse.

This will make it more difficult to converge and increase the chance that we have false stationarity by being trapped on local optima for longer.

We can (as I've mentioned earlier) treat the model as a nuisance variable and propose model jumps in the MCMC (via rjMCMC), and this makes the space even more complex.

The complexity of the space can be dealt with to some degree by running several chains simultaneously (Metropolis-Coupled MCMC, or MC^3), and every now and then proposing a state from a different chain and trying to switch states in the cold chain for states the other chains.

Typically, the Metropolis-coupled chains are proposing more radical proposals. This is called heating the chains, because we're trying larger steps in the heated chain and every now and then, trying to update the cold chain. This actually permits traversal away from local optima.

In addition, we have the problem of **truncating priors**.

Say we have a parameter such as the gamma distribution shape parameter, which is unbounded.

If we want to use a flat prior for this parameter, we need to truncate it. In the example above, it's truncated at the lower end by 0.05 and at the upper end by 50. This certainly seems reasonable, but it creates a bias.

Joe gives an example on page 305 where different truncations of a flat prior on a branch length leads to Bayesian estimate that excludes the ML estimate.