Appendices to STACEY package documentation

Graham Jones, www.indriid.com

2015-11-30, October 7, 2022

STACEY version 1.3.0

Contents

1 Priors

The BEAST book (**?**) contains some good advice about priors. I'll assume you've read it. I'd like to add a few more bits of advice: some general points, then some specific to parameters in STACEY.

1.1 Quantiles

In choosing a prior, it is often better to look at the quantiles of the distribution than at the mean and variance. One reason is that it is usually easier to express biological or other expert knowledge in terms of quantiles. Another reason for this is that most of the parameters in phylogenetics are positive real numbers, and highly skewed distributions are often appropriate for a prior. The mean and variance do not summarise such a distribution very well.

1.2 Lognormal, gamma and inverse gamma distributions

These three distributions are available in BEAST and all have support $[0, \infty)$. The shape of the lognormal is normal in log space, and is 'between' the other two. In comparison, the gamma distribution gives a higher probability to very small values, and has a sharper cutoff for very large values. The inverse gamma is the other way around, with a sharper cutoff for very small values, and a higher probability assigned to very large values. These tendencies become more pronounced as the distributions become more diffuse.

Figure 1 shows a lognormal with mean zero and standard deviation 4 in log space (`meanlog=0`, `sdlog=4` in R, M=0, S=4 in Beauti), a gamma with shape parameter 0.25 and a scale parameter of 22.897, and an inverse gamma with shape parameter 0.25 and a scale parameter of 1/22.897. All three distributions have median 1, or 0 in log space. They are shown in log space because when plotted in 'real' space, they all look like a spike at 0, up the y-axis, and a tail along the x-axis. In log space the gamma and inverse gamma are 'reflections' of one another. The 95% intervals for these lognormal, gamma, and inverse gamma distributions are approximately [0.0004, 2500], [0.000006,39], and [0.025, 170000] respectively. In all three cases their 95% intervals cover a range of about 6.5 million.

1.3 Priors, models, and data

Theory ensures that if the model is correct, then any prior that assigns a nonzero probability to the true value will be overwhelmed by enough data. But models are always approximations, and as the amount of data available increases, the tendency is to use more complex models with more parameters to be estimated. Then there isn't enough data again. Furthermore, while there may be plenty of data to estimate most parameters, there may be little signal for others. So in practice, you can rarely rely on the theory, and badly chosen priors can undermine an analysis. For the most part, models are much more important than priors, but you need to be alert to situations when they are not.

The choice between simple and complex models is often described as bias-variance trade-off. The simple model will be biased, but at least you will be able to estimate its parameters well. The complex model will be less biased, because it is closer to reality, but its parameters may be poorly estimated. Bayesian priors can offer a compromise: use the complex model but use concentrated priors for some or all of the extra parameters. In my opinion, this should be done more often than it is.

1.4 Suggested approach

Here is suggested method for a subjective or informative prior. This could be based on expert opinion, or could be 'borrowed' from results of other analyses on similar data. Suppose θ is a real-valued parameter.

Start with the median, and choose a value x so that it seems about equally likely that θ is bigger or smaller than x. Then you could consider the 0.05 and 0.95 quantiles. For the 0.05 quantile, you are trying to estimate an x so that

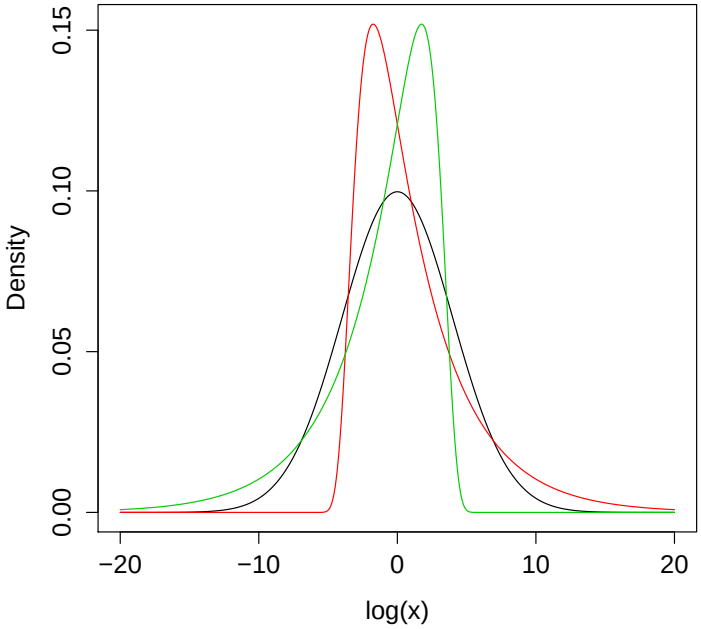

Figure 1: Very diffuse distributions. Lognormal (black), gamma (green), and inverse gamma (red) distributions in log space.

in your opinion, $\Pr(\theta < x) = 0.05$. You may need to experiment with different distributions and their parameters to get what you want. There is little point in getting more than a rough match.

Finally consider some extreme but still possible values for x. Expert opinion can be wrong, and apparently similar analyses and data sets may turn out to be quite different. You don't want to rule out things that might actually happen, and you may need to make the prior more diffuse to achieve this. In general it is better to err on the side of being too diffuse than of being too concentrated. On the other hand, they should not be absurdly diffuse. Even if you have 'no idea' what size a parameter might be, you will very rarely need distributions that are as diffuse as the ones described in 1.2.

If you want to pursue this sort of approach further, **?** is a good start.

1.5 Yule and birth-death models

In STACEY (among many other phylogenetic models) the species tree is assumed to be the result of a birth-death process. A special case is the pure-birth, or Yule process. Denote the birth rate (speciation rate) by λ and the death rate (extinction rate) by μ. Assume the units are expected substitutions, which is the usual case for STACEY. If there is one species to begin with, then after t expected substitutions, the expected number of species is $\exp((\lambda - \mu)t)$. The value $\lambda - \mu$ is known as the growth rate or diversification rate, called `bdcGrowthRate.t:Species` in the XML. There is a formula in the BEAST book (7.2.3, p109) for the case $\mu = 0$, and a known number of species. For choosing a prior, we don't need this precision. Roughly, the growth rate is about 1 divided by a typical branch length. So if you expect the branches in your species tree to be around 0.01, you expect $\lambda - \mu$ to be very roughly 100. In this case, you might use a lognormal with M=4.6, S=1 for the growth rate. This is a fairly diffuse prior with median $e^{4.6} \approx 100$ and a 95% interval (in log space) which is about [14.0, 700]. If you were less sure about the branch lengths, M=4.6, S=2 would give the same median but a 95% interval of approximately [2.0, 5000].

The other parameter is the relative death rate μ/λ, called `relativeDeathRate.t:Species`. In general, this is much harder to estimate than the growth rate. Deciding whether to estimate it is a dilemma of the type described in section 1.3. Some evidence suggests it is usually about 0.65 (**?**), or 0.9 (**?**). Often in phylogenetic analyses, we are looking at groups of species which have diversified a lot recently, so a lower value might be appropriate, though it is hard to tell if a group of species has done well by chance, or because of some intrinsic property. If you want a simple model, then, in my opinion, fixing $\mu/\lambda = 0.5$ is preferable to fixing $\mu/\lambda = 0.0$ (the Yule model) for most analyses. For larger numbers of species, it may be better to estimate μ/λ.

1.6 Population size parameters

The prior for the population size parameters is made of two parts. Firstly, there is `popPriorScale` which is the overall scaling factor for the population size parameters. It is denoted as σ in the STACEY paper. The actual effective population size parameters for the SMC-tree branches are then drawn from another distribution and multiplied by `popPriorScale`. The second distribution is a mixture of inverse gammas.

To minimise confusion I recommend using a mixture of inverse gammas which is centred around 1. The default is a single inverse gamma with mean 1. You can use more diffuse inverse gammas by decreasing the α parameter, but the mean becomes a poor way to characterise the distribution as α decreases, and becomes infinite once $\alpha \leq 1$. It would be better to keep the median near 1 if you want to use small α values.

Given this condition on the mixture of inverse gammas, the `popPriorScale` is approximately the average over branches of $N\mu$ where N is the effective population size along a branch, and μ is the mutation rate in substitutions per site per generation in the branch. For a single diploid population, the expected time (going back from the present) for a coalescence to occur betwen a pair of gene copies is $2N$ generations, so the expected TMRCA for this pair is $2N\mu$ substitutions, and the expected genetic distance between a pair is $4N\mu$ substitutions. You can use this as a guide to choose a prior for `popPriorScale`.

Now we turn to the second stage of the prior, which represents the branch-to branch variation among the population size parameters. The manual shows the density for the default prior. I don't think anyone knows whether this represents a typical pattern of variation of population sizes over branches. I suspect most people will stick to the default because they don't know what else to do. However I'm sure that the default is inappropriate for *some* analyses. An alternative prior, which is more like a lognormal, and more diffuse than the default, can be achieved using this XML:

```
<popPriorInvGamma id="InvGammaComponent.1" spec="stacey.InverseGammaComponent">
<parameter id="InvGammaComponentWeight.1" estimate="false" name="weight">1.0</parameter>
<parameter id="InvGammaComponentAlpha.1" estimate="false" name="alpha">1.5</parameter>
<parameter id="InvGammaComponentBeta.1" estimate="false" name="beta">0.31348</parameter>
</popPriorInvGamma>
 <popPriorInvGamma id="InvGammaComponent.2" spec="stacey.InverseGammaComponent">
<parameter id="InvGammaComponentWeight.2" estimate="false" name="weight">1.0</parameter>
<parameter id="InvGammaComponentAlpha.2" estimate="false" name="alpha">2.0</parameter>
<parameter id="InvGammaComponentBeta.2" estimate="false" name="beta">0.94044</parameter>
</popPriorInvGamma>
<popPriorInvGamma id="InvGammaComponent.3" spec="stacey.InverseGammaComponent">
<parameter id="InvGammaComponentWeight.3" estimate="false" name="weight">1.0</parameter>
<parameter id="InvGammaComponentAlpha.3" estimate="false" name="alpha">2.5</parameter>
<parameter id="InvGammaComponentBeta.3" estimate="false" name="beta">2.8213</parameter>
</popPriorInvGamma>
<popPriorInvGamma id="InvGammaComponent.4" spec="stacey.InverseGammaComponent">
<parameter id="InvGammaComponentWeight.4" estimate="false" name="weight">1.0</parameter>
<parameter id="InvGammaComponentAlpha.4" estimate="false" name="alpha">3.0</parameter>
<parameter id="InvGammaComponentBeta.4" estimate="false" name="beta">8.4640</parameter>
</popPriorInvGamma>
```

Table 1 shows quantiles for the two distributions. The 'Single' distribution is the same as the default, except it has

a scale β of 2.674 instead of 2.0 in order to make its median equal to 1 instead of its mean equal to 1.

Quantile	Single, $\alpha = 3.0$	Mixture of four
0.0005	0.222	0.042
0.0050	0.288	0.064
0.0250	0.370	0.100
0.2500	0.682	0.400
0.5000	1.000	1.000
0.7500	1.548	2.353
0.9750	4.322	8.707
0.9950	7.915	17.195
0.9995	17.862	44.024

Table 1: Quantiles for two mixtures of inverse gammas.

1.7 Collapse weight

The prior for the collapse weight ω (`collapseWeight.t:Species` in the XML) can be used to provide prior information about the likely number of species in a delimitation analysis. This prior is usually a beta distribution, with parameters a and b (`alpha.bdcCollapseWeight.smcTreePrior` and `beta.bdcCollapseWeight.smcTreePrior` in the XML). Assume there are n minimal clusters. Then the prior probability of the number of species k for $1 \leq k \leq n$ follow a distribution given by

$$\Pr(k = x|n, a, b) = \frac{\Gamma(n)}{\Gamma(x)\Gamma(n+1-x)} \frac{\Gamma(x-1+b)\Gamma(n-x+a)}{\Gamma(n-1+a+b)} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}.$$

This is a beta-binomial distribution, shifted by 1. The R function `px` below calculates this, for $1 \leq x \leq n$.

```
library(VGAM)
px <- function(x, n, a, b) {
dbetabinom.ab(x-1, size=n-1, shape1=b, shape2=a)
}
# example of use
z <- px(1:25, 25, 8,2)
plot(z)
print(z)
```

1.8 Finally

Do not follow this advice, or any other advice about priors, without thinking. I don't know anything about your organisms, your data set, or your purpose in doing the analysis. Nor do the developers of any other phylogenetic software. In a Bayesian analysis, priors are your responsibility in the same way that your choice of model is your responsibility.

2 Operators, convergence, and mixing

This section will mainly be of interest to users with analyses that are taking a very long time to obtain good ESSs. Some preliminary results using STACEY on large data sets are at
`http://www.indriid.com/2015/2015-08-31-bigdata.pdf` and
`http://www.indriid.com/2015/2015-11-01-morebigdata.pdf`.

STACEY has five 'big' operators `NodesNudge`, `FocusedNodeHeightScaler`, `CoordinatedPruneRegraft`, `StaceyNodeReheight`, and `ThreeBranchAdjuster` which change the shape of the SMC-tree. There is also an overall 'updown' operator which squashes or stretches all trees and related parameters, and others which change the population scaling factor and the parameters associated with birth-death-collapse model (growth, relative death, collapse weight, origin height) for the SMC-tree. I will call these 'SMC-tree operators', and the five named ones the 'big SMC-tree operators'. The first three of these are described in **?**. Here are some details about the other two.

2.1 StaceyNodeReheight

The acceptance rate of the standard *BEAST NodeReheight operator decereases as the number n of loci increases. For large numbers of loci, the acceptance rate appears to be approximately $\propto 1/n$. This is not surprising, since the height gap between the SMC-tree node and the gene tree node which limits its movement becomes smaller with more loci.

In STACEY 1.2.3, new heights are sampled with a bias toward the maximum compatible height h. The density used is proportional to $1/(h+s-x)$ where s is an small positive tuning parameter. The support of the distribution is $[0,h]$, the same as for the standard version. The CDF of the distribution is

$$F(x) = \frac{1}{\log(h+s)-\log(s)}(\log(h+s)-\log(h+s-x))$$

The PDF is

$$f(x) = \frac{1}{\log(h+s)-\log(s)}\frac{1}{h+s-x}$$

and the inverse CDF is

$$G(y) = h+s-\exp(\log(h+s)(1-y)+\log(s)y).$$

The inverse CDF $G()$ is used for the new sample and the PDF $f()$ is used for calculating the Hastings ratio. By default s is set to $0.1\sigma/n$ where σ is the population scaling factor. The idea is that σ/n is approximately proportional to the height gap mentioned above. Smaller values of s make the density more concentrated near h so more likely to be accepted. The median is $G(1/2) = h+s-\sqrt{s(h+s)}$, which can provide a guide to choosing s.

`StaceyNodeReheight` has an attribute `proportionUniform` which controls the amount of time the operator uses a uniform distribution like StarBEAST. This defaults to 0.1. Use `proportionUniform="1.0"` to sample uniformly all the time (ie turn off the new behaviour).

`StaceyNodeReheight` also has an attribute `tuning` which controls s. This defaults to 1.0. for example, use `tuning="5.0"` to make s five times as big. In the case of species delimitation, where many branches are tiny, changing s may have little effect on the acceptance ratio.

2.2 ThreeBranchAdjuster

This is an implementation of the 'rubber-band' operator of **?**. It changes the height of one SMC-tree node, and the heights of all gene tree nodes within the the three adjacent branches. No topologies are changed, and all gene tree nodes stay within the same SMC-tree branches. A new height h' for the SMC-tree node is sampled uniformly in $[h-s, h+s]$ where h is the old node height and s is a small positive tuning parameter. (If h' falls outside the allowed range, it is 'reflected' from the limit.) The default value of s is the minimum of $40\sigma/n$ and half of the allowed range. The dependence on σ/n is for a similar reason to that described for `StaceyNodeReheight`.

`ThreeBranchAdjuster` has an attribute `tuning` which controls s. This defaults to 1.0. for example, use `tuning="5.0"` to make s five times as big. In the case of species delimitation, where many branches are tiny, changing s may have little effect on the acceptance ratio.

2.3 Convergence, pseudo-convergence, and mixing

> At the moment the choice of a good proposal distribution [operator] involves the burning of incense, casting of chicken bones, use of magical incantations, and invoking the opinions of more prestigious colleagues. (**?**, p296.)

I do not think things are that bad now (if they ever were), but the design of good operators is still a matter of art or engineering as much as science. We do not have a good understanding of how well this or that choice of operators is going to sample from the posterior in complex areas like phylogenetics.

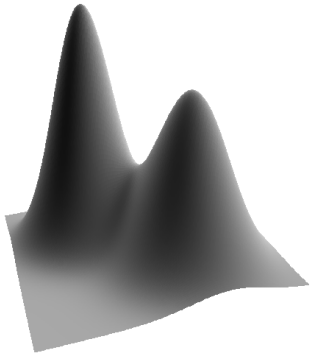

Figure 2: Multimodal

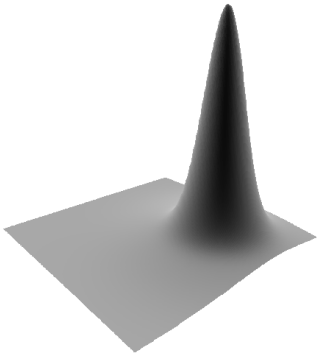

Figure 3: Witch's hat

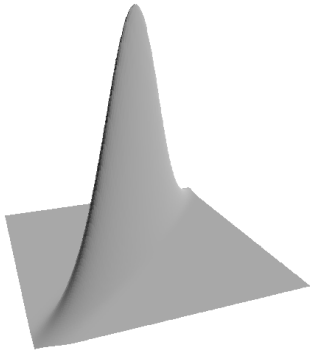

Figure 4: Ridge

It can be extremely difficult to distinguish genuine convergence from pseudo-convergence (**?**, 1.11.2.) If your chain has only pseudo-converged your results are dubious at best. Pseudo-convergence can occur when the posterior is multimodal (Fig 2). In this example, both peaks have the same volume - the shorter one is wider - so the MCMC should sample from both equally. If the chain gets stuck on one peak for a very long time it will look like successful convergence. Pseudo-convergence can also occur when the posterior is a mixture of a sharp peak and a broad one, as in the 'witch's hat' shape in Fig 3. Again, the two components have equal volume. If the chain is making big jumps, it will stay in the broad peak for ages, not 'noticing' the sharp peak. When it does, it will jump there but get stuck at the same point for ages. If the chain is making small jumps, it will stay in the sharp peak for ages before it wanders down to explore the broad peak, which it will do very slowly before returning to the sharp peak. Although the posterior is unimodal, it can still give a false appearance of convergence.

Fig 4 shows another tricky case. Suppose the only operators available move parallel to the axes (the sides of the base square). Here the difficulty is staying on the ridge. Unless the the move is very small, it will 'fall off' the ridge on one side or the other. The result is that the chain zigzags extremely slowly along the ridge. I don't think this is pseudo-convergence exactly, but it can be hard to detect the problem. In extreme cases, it may appear that the chain is mixing well, when in fact it is just stuck near the first point on the ridge which it happened to reach.

It may seem that the cases in Figs 3 and 4 are rather pathological and unlikely to occur in real analyses. If the posterior was only two-dimensional, I would agree, but the posteriors explored in phylogenetics can be very high-dimensional, where such things are far more likely. In particular, each node height of the SMC-tree and the gene trees is a different dimension, and I think this leads to many 'ridge' situations. A 'diagonal' operator which made moves in approximately the direction of the ridge would help a lot in the two dimensional case of the picture. In the much higher dimensional case of the multispecies coalescent posterior, it is much harder to find a good direction. `NodesNudge`, `FocusedNodeHeightScaler`, and `ThreeBranchAdjuster` are three attempts to find good directions to move in this space.

Assuming the chain has genuinely converged, the next question is how quickly is it exploring the posterior. It takes a huge amount of computer time to obtain good estimates of ESSs/hour for large data sets. Different random seeds for the MCMC algorithm may give different results. Different replicates from simulated data may behave quite differently. The ESSs may be good for some parameters and poor for others. There is thus a huge amount of variance which impedes the search for patterns. See **?** for some work on this. Real data almost always violates the model in some way, and this complicates the picture further.

2.4 Speedups and pseudo-convergence

In my opinion some methods which attempt to speed things up are likely to increase the risk of pseudo-convergence. I think that any attempt to 'get through burnin' quickly which is not part of the normal MCMC algorithm is risky. If the MCMC algorithm cannot get through burnin quickly enough, it seems likely that it cannot explore the posterior thoroughly enough either. For this reason I have avoided the 'SBI' initialisation method which is implemented in BEAST2 for StarBEAST. SBI estimates the initial gene trees using UPGMA, then chooses an initial species tree based on these. Since UPGMA is known to be biased, it seem to me that all initial states will suffer from the same bias. This shouldn't matter, because the MCMC algorithm will eventually sample fairly from the posterior from any initial state. However, I am concerned that with some data sets, such an initialisation will reliably push the chain towards a particular region where it might stay for so long that it appears to be converged, but has not. STACEY uses an initialisation like StarBEAST in BEAST1: it uses a random SMC-tree, and gene trees which are forced to have all nodes in the root branch of the SMC-tree but are otherwise random.

Secondly, I am wary of self-tuning operators. I think that in 'witch's hat' situations these may backfire. The operators will become tuned to either the sharp or broad component, and become very bad at exploring the other. For some of the operators that seem most important, I have used a range of fixed (not optimised) operators. the snippet below is abbreviated from the XML.

```
<operator id="xScaler100.t:Species" optimise="false" scaleFactor="0.100" ... />
...
<operator id="xScaler999.t:Species" optimise="false" scaleFactor="0.999" ... />
```

I emphasize that these concerns are speculation on my part, and I do not have examples of things going wrong with SBI or auto-optimising operators. Nor is there any guarantee that the initialisation method used by STACEY is better. It seems 'more random' to me, since different seeds give different random starting topologies for all the trees. On the other hand, forcing all the gene tree nodes into the SMC-tree's root is not ideal.

2.5 Operator weights

All the SMC-tree operators have IDs ending in 'Species', and by default they are assigned a total weight of 20% by Beauti. The remaining 80% is assigned to gene tree operators, and their associated parameters for clock models, site models, etc. By default, the 20% is assigned mainly to the big SMC-tree operators, so they get a bit less than 4% each. This may not be appropriate. To change the values in Beauti, untick `Automatic change operator weights for *BEAST analyses` in the `Mode` menu. Then open the operators panel via the `View` menu.

For large data sets, it may be worth experimenting with operator weights (these and others) in order to improve the rate at which ESSs/hour can be obtained. The big SMC-tree operators take a longer time than most. All but `StaceyNodeReheight` change the gene trees forcing a recalculation of the gene tree likelihoods, so they are slower than `StaceyNodeReheight`. For this reason I suggest giving `StaceyNodeReheight` about three times the weight of the other four. As a tentative suggestion, here is a table for these operators, as percentages of the total for all the operators.

Number of loci	Total for big SMC-tree ops	StaceyNodeReheight	Other four
1	34	14	5
3	22	10	3
10	14	6	2
30	8	3.6	1.2
100	5	2.0	0.7
300	2.8	1.2	0.4
1000	2.1	0.9	0.3

As another rough rule of thumb, I suggest that the big SMC-tree operators should account for about a half of the total time. You can set their weights to zero, run a few million generations and note the time per million samples. Then increase their weights until the time is approximately double this.

2.6 BEAGLE

I have not found BEAGLE reduces the runtime for STACEY. This probably depends on the data and on the GPU. BEAGLE does not speed up the calculation of the multispecies coalescent, which takes a substantial fraction of the time, so a really dramatic speedup with BEAGLE is not possible.