Model Stacking in Bayesian Phylogenetic Inference

Liyi Zhang, Yuling Yao, Andrew Gelman

November 19, 2020

Methodology, Model, and Notations

Goal
Given information of leaf nodes, infer tree structure behind the leaf nodes. Tasks include finding or approximating posterior distributions of parameters, model selection, and prediction.

Notations
$\mathcal{Y}$ denotes observed data that is cleaned (transformed from ACGT to matrices of 0 and 1). $\mathcal{T}$ denotes tree *topology*, which refers to parent-child relationships in a tree's node structure. λ denotes branch lengths on the tree, and can be considered as evolution times. θ denotes other continuous parameters such as the rate matrix Q.

Model
The posterior of parameters is expressed as:

$$p(\mathcal{T}, \lambda, \theta|\mathcal{Y}) = \frac{p(\mathcal{Y}|\mathcal{T}, \lambda, \theta)p(\mathcal{T}, \lambda, \theta)}{p(\mathcal{Y})}$$

In our implementations, we randomly select a tree topology, and compute posterior of branch lengths and other continuous parameters given observed data. In the current Stan program, the first line of model block draws branch lengths, denoted λ here, from an Exponential prior, the second line draws another parameter, denoted κ here, from an Exponential prior, and the third line draws data from $p(\mathcal{Y}|\mathcal{T}, \lambda, \theta)$, computed using Felsenstein's pruning algorithm.

Thus, we essentially have K models. The j-th model is defined by the following priors and likelihood:

1. $$\lambda \sim Exponential(l) \qquad \kappa \sim Exponential(k)$$

2. $$\mathcal{T}_j \sim P_j(\mathcal{T})$$

   where $P_j(\mathcal{T} = \mathcal{T}_j) = 1$.

3. $$\mathcal{Y} \sim p(\mathcal{Y}|t, \lambda, \theta)$$

   where p follows Felsensetein's algorithm discussed below.

Likelihood details: $p(\mathcal{Y}|\mathcal{T}, \lambda, \theta)$

We follow these steps as we calculate the likelihood given parameters. It follows Felsenstein's pruning algorithm, used in Mau et al. 1999 and various later versions of Bayesian phylogenetic inference. [Justification can be written later]

1. Construct rate matrix Q. Q must satisfy the following properties: 1) its dimension is a-by-a, where a is number of states, and $a = 4$ if ACGT are used; 2) non-diagonal entries are positive; 3) rows sum to 0.

   Q is paramterized with parameter κ: (The magnitude of Q will be governed by λ when it is used.)

   $$\begin{pmatrix} - & 0.25 & 0.25\kappa & 0.25 \\ 0.25 & - & 0.25 & 0.25\kappa \\ 0.25\kappa & 0.25 & - & 0.25 \\ 0.25 & 0.25\kappa & 0.25 & - \end{pmatrix}$$

   where hyphens '-' refer to negative sum of other entries in the same row.

2. Calculate transition matrix $P(\lambda) = exp(Q \cdot \lambda)$. exp is matrix-exponential, and λ is branch length (scaler) for a particular branch on the tree.

3. Given a tree topology, for each node is defined a matrix M, and root node's matrix is used to calculate $p(\mathcal{Y}|\mathcal{T}, \lambda, \theta)$. M has dimension s-by-a, where s is number of sites (length of ACGT data). We first write likelihood for each leaf node. For example, we denote A as $[1, 0, 0, 0]$, C as $[0, 1, 0, 0]$, G as $[0, 0, 1, 0]$, T as $[0, 0, 0, 1]$, and if a leaf node has ACC, then this node's matrix M is:

   $$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

   For each parent, its matrix is computed as:

   $$M_{parent} = (M_{left} \cdot P(\lambda_{left})) * (M_{right} \cdot P(\lambda_{right}))$$

   where $P(\lambda)$ is defined above, and * is elementwise multiplication. The result of the above operation is still s-by-a.

4. Define a state-probability vector, which is set as $\pi = (1/a, ..., 1/a)^T$.

   $$p(\mathcal{Y}|\mathcal{T}, \lambda, \theta) = \prod_{i=1}^{s} (M_{root} \cdot \pi)_i$$

Model Selection with Stacking

We refer to *Yao,et al. (2017) - Using stacking to average Bayesian predictive distributions* - to conduct model combination using stacking of predictive distributions. The results can be compared with that of *Yang and Zhu (2018)*, who examined Bayesian Model Averaging on phylogenetic tree models. We denote a model by M_k.

We empirically approximate the full predictive distribution $p(\widetilde{y}|y, M_k)$ by leave-one-out predicative distribution. In leave-one-out terminologies, each data-point is considered as a site in the ACGT sequence of each leaf-node. For example, if we leave out the s-th data-point, we compute the likelihood of only the s-th position of the ACGT sequence at each leaf-node. Following the stacking of predicative distributions, we maximize an objective over weights assigned to the models:

$$\max_{w} \frac{1}{S} \sum_{s=1}^{S} log \sum_{k=1}^{K} w_k p(Y_s | Y_{-s}, M_k)$$

Weights vector w sums up to 1 and has non-negative entries.

Data Analysis and Visualization

MCMC Analysis

We fit models on a data consisted of 3 leaf-nodes, each 100-sites-long. Sites are randomly generated for our first model, with equal probabilities for A, C, G, and T. We use the following sequential process to analyze model fits:

- Randomly generate data
- Arbitrarily select a topology $\mathcal{T}$
- Repeat
  - Fit model using Stan on the newly generated data and the selected topology $\mathcal{T}$
  - Generate data using parameter posteriors from the above model fit

I. Single Model Diagnostics

We first examine MCMC diagnostics for a single model. We repeated the above process five times to generate five model fits. The program reports no divergent chains, and Figure 1 shows the traceplots of generated parameters for the first model fit. The sequences of different chains have mixed and are stationary. However, the posteriors for lambda and kappa have high variance, as shown by both the high, protruding values in Figure 1, and an inspection at the data provided by Stan's summary of the model fit.

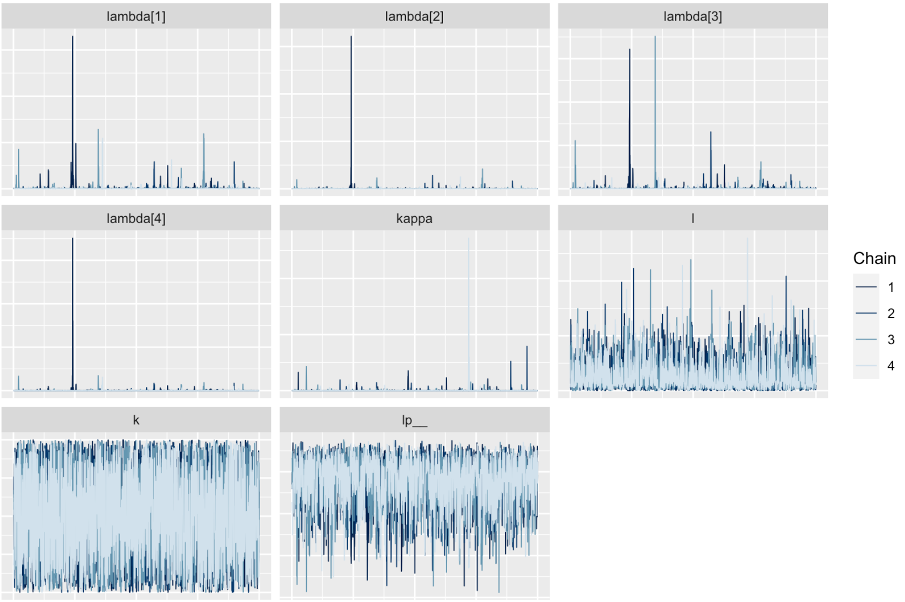
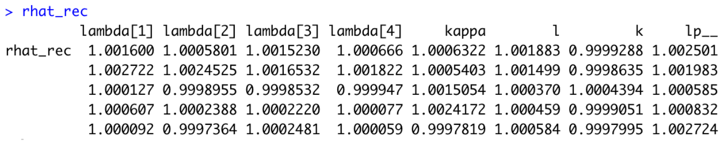
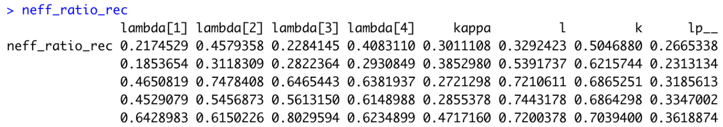


Figure 1: Traceplots for the first model fit. Each lambda denotes a branch length, kappa is a term on Q-matrix (heuristically, how much more likely for A to become G than to become C or T), l is parameter for lambda's prior, which is Exponential, and k is parameter for kappa's prior, which is also Exponential.



```
> rhat_rec
         lambda[1] lambda[2] lambda[3] lambda[4]     kappa        l         k     lp__
rhat_rec  1.001600 1.0005801 1.0015230  1.000666 1.0006322 1.001883 0.9999288 1.002501
          1.002722 1.0024525 1.0016532  1.001822 1.0005403 1.001499 0.9998635 1.001983
          1.000127 0.9998955 0.9998532  0.999947 1.0015054 1.000370 1.0004394 1.000585
          1.000607 1.0002388 1.0002220  1.000077 1.0024172 1.000459 0.9999051 1.000832
          1.000092 0.9997364 1.0002481  1.000059 0.9997819 1.000584 0.9997995 1.002724
```

Figure 2: R-hat for five model fits, each row a model fit.



```
> neff_ratio_rec
               lambda[1] lambda[2] lambda[3] lambda[4]     kappa         l         k      lp__
neff_ratio_rec 0.2174529 0.4579358 0.2284145 0.4083110 0.3011108 0.3292423 0.5046880 0.2665338
               0.1853654 0.3118309 0.2822364 0.2930849 0.3852980 0.5391737 0.6215744 0.2313134
               0.4650819 0.7478408 0.6465443 0.6381937 0.2721298 0.7210611 0.6865251 0.3185613
               0.4529079 0.5456873 0.5613150 0.6148988 0.2855378 0.7443178 0.6864298 0.3347002
               0.6428983 0.6150226 0.8029594 0.6234899 0.4717160 0.7200378 0.7039400 0.3618874
```

Figure 3: Neff-ratio for five model fits, each row a model fit.

R-hat and Neff-ratio (number of effective draws over total draws) are shown in Figure 2 and 3, respectively. They are shown for all five model fits, and each row is one model fit. These values raise no suspicion on the model so far.

II. Across-Model Diagnostics

For some reason, posterior variance does drop for the second through fifth model fits. High posterior variance indicates that the model is 'unsure' of the branch length, resulting in a distribution with high entropy. An explanation is that the arbitrarily selected topology $\mathcal{T}$ is an unlikely topology, so the model concludes with a high posterior variance. Later models generate data based on the previous model's knowledge of the parameters, so the arbitrarily selected topology $\mathcal{T}$ becomes a likely topology for later models' data. Then, posterior variance decreases.

Figure 4 is also a traceplot, but it concatenates the second and third model's traceplots. The division between the second and third model's result would be an imagined vertical line right in the middle of the x-axis. Once again, the sequences of different chains have mixed and are stationary (viewed in terms of each individual model), but posterior variance has decreased. Across the two models, distributions of kappa, as well as k, are similar, but lambdas have different distributions. This phenomenon is actually logical given our experimental setup: the latter model is more 'confident' of the arbitrarily selected topology $\mathcal{T}$. Since lambdas signify branch lengths, or evolutionary time, more 'confident' models infer lambdas that are centered on smaller values.

III. Model Diagnostics for Different Topologies

We also modify the sequential algorithm that we have been doing previously. We want to test the previous explanation for the problem: why does model 1 exhibit high posterior variance. We mentioned that the topology being fitted

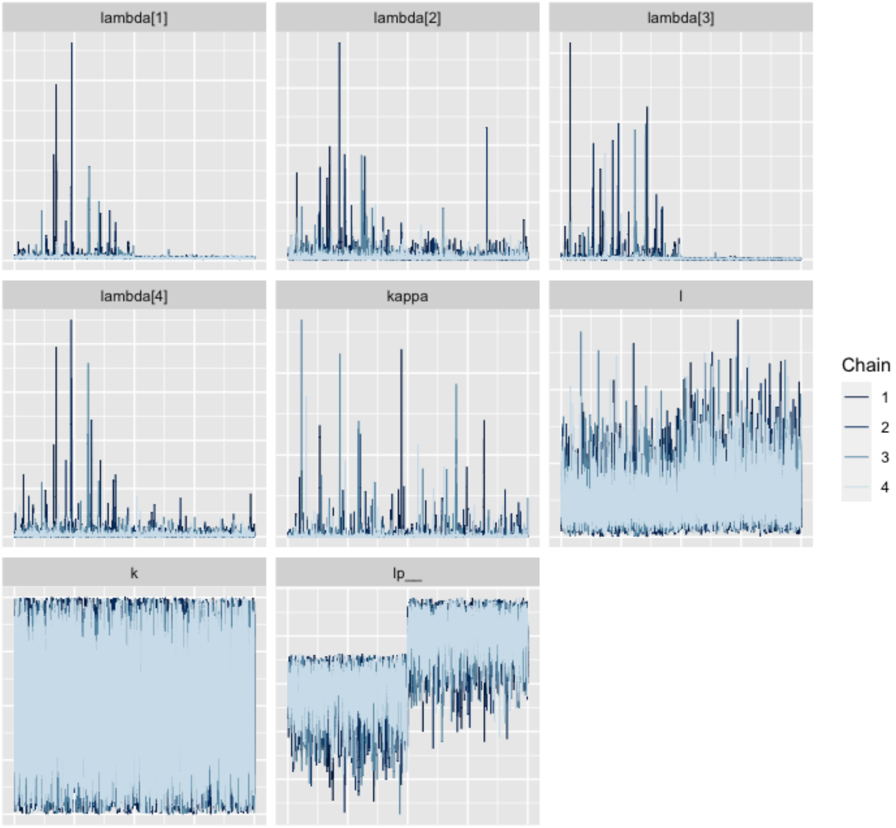


Figure 4: Traceplots for the second and third model fit. Notations are same as Figure 1, but this traceplot concatenates two model fits. On each plot, the left half refers to model 2, and the right half refers to model 3.

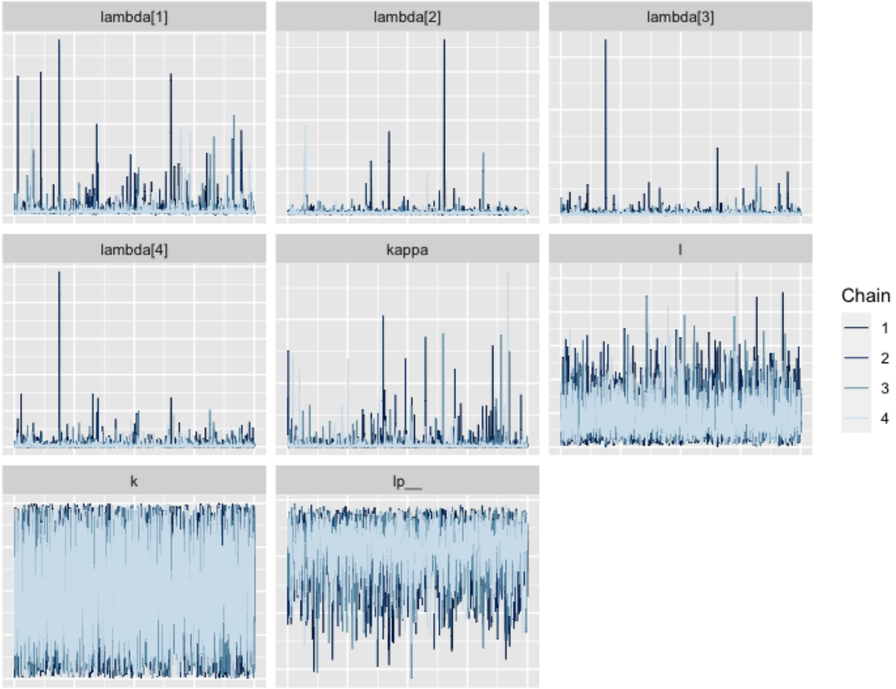


Figure 5: Traceplots for the model that, among the three models that fit three topologies, exhibits lowest posterior variance. Notations are same as Figure 1.

previously may happen to be the unlikely topology. Therefore, on random data of the same size, we fit models based on all three topologies. Posterior variance indeed varies a lot across these three models. Figure 5 shows traceplot for the model exhibiting the lowest posterior variance. Again, chains appear mixing and stationary, but the visualization is clearer here than in Figure 1: the protruding high values of lambda are not as fierce, so we can visualize the main bulk of lambda values that are low.

IV. Model Diagnostics by Recovering Parameters

We generate data with an arbitrary set of parameters, and use MCMC to fit this data and compare the posterior samples with the original set of parameters. We use parameter set $\theta = \{\lambda = (1, 1, 1, 1, 1, 1), \kappa = 2\}$ (Kimura two-parameter model, Wang 2012), and four leaf-nodes, each of length 1000.

Figure 6 and 7 show both the original set and posterior samples of parameters. The figures suggest that the original set of parameters can be recovered. It is also worthwhile to note that the length of data in each node, $S = 1000$, was deliberately chosen. Experiment on $S = 100$ would yield posterior samples that have too much variance.

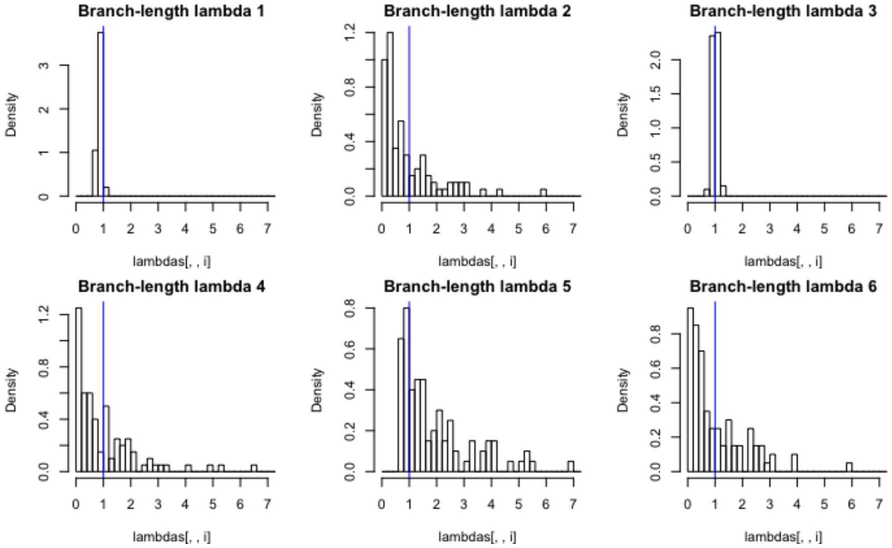


Figure 6: Histogram of posterior samples of branch-lengths (3 left-branches, 3 right-branches). The original parameter that was used to generate data is shown as the blue line.

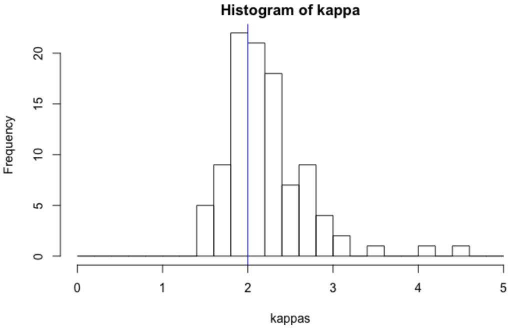


Figure 7: Histogram of posterior samples of kappa parameter. The original parameter that was used to generate data is shown as the blue line.

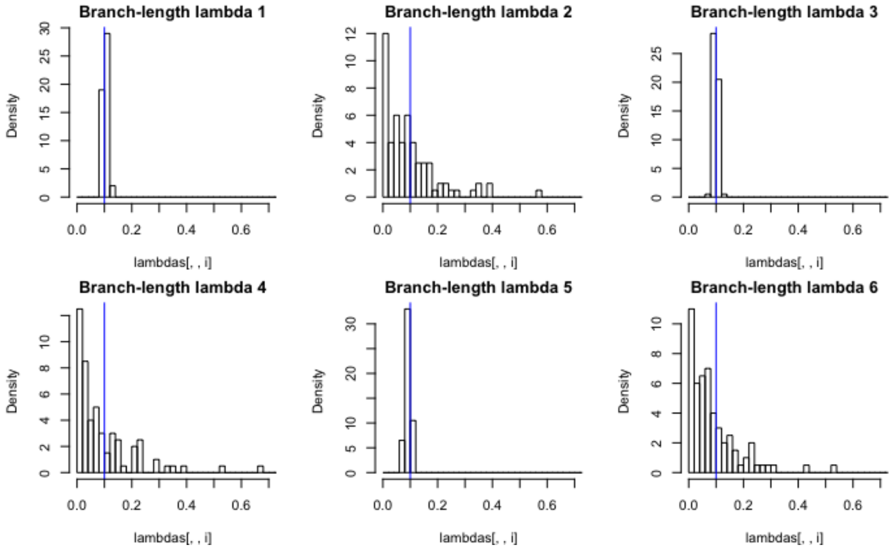


Figure 8: Histogram of posterior samples of branch-lengths (3 left-branches, 3 right-branches). The original parameter that was used to generate data is shown as the blue line.

A similar experiment was done, but with branch-lengths $\lambda = 0.1$. Figure 8 and 9 show both the original set and posterior samples of parameters. While λ is well-recovered, κ is not.

Stacking Analysis

I. Simulation - Stacking and Bayesian Model Averaging (BMA)

Stacking. We fit models on a data consisted of 3 leaf-nodes, each 100-sites-long. Sites are randomly generated, with equal probabilities for A, C, G, and T. Since there are three distinct topologies, three models are fitted. The above process was repeated 100 times, where each time sites are randomly generated again from scratch. We compute and record the weights vector w based on the stacking objective function aforementioned. There will reasonably be 100 weights vectors, each of length 3. Figure 10 plots w.

BMA. We also processed BMA on the same 100 groups of randomly generated data, where we computed the probability of model given data:

$$P(M_k|Y) \propto p(Y|M_k)P(M_k)$$

The evidence $p(Y|M_k)$ is computed using adaptive path sampling (Yao, et al., 2020). The prior of each model is such that $P(M_k) = \frac{1}{K}$. Figure 11 plots 100 vectors, where each vector is a model posterior: $(P(M_1|Y), P(M_2|Y), P(M_3|Y))^T$.

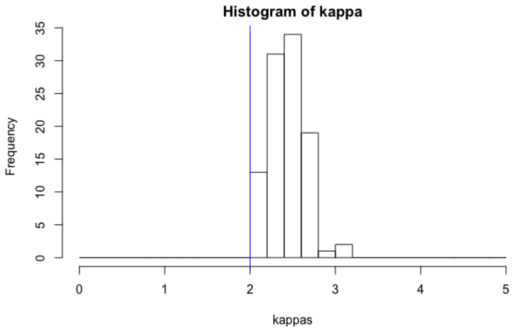


Figure 9: Histogram of posterior samples of kappa parameter. The original parameter that was used to generate data is shown as the blue line.

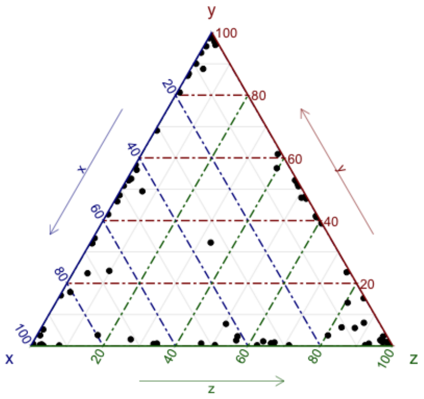


Figure 10: Distribution of weights vector w. Labels are percentages, and, for example, the a point at the top vertex refers to (0, 1, 0), and a point in the middle is about (1/3, 1/3, 1/3).

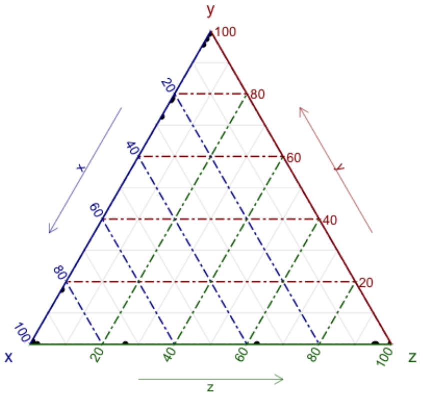


Figure 11: Distribution of BMA $P(M_k|Y)$.

Compared with the plot of stacking weights, the plot for BMA shows more cases where one model is given the assignment of almost 1.

II. Simulation - Stacking for Prediction

We run the model on a short section of real data consisted of 4 leaf-nodes, each about 30-sites-long. Then, we compute posterior predicative distribution for two extra sites on the first leaf-node, assuming knowledge of these two extra sites on the other three leaf-nodes. Since each site in our data refers to ACGT, this distribution is a discrete distribution with a domain cardinality of 4.

We use genome sequences of Human, Gibbon, Aaardvark and Armadillo, and have knowledge of ground-truth topology. (In fact, there is debate regarding the ground-truth topology, and here we are using interpretations found on American Museum of Natural History that are based on genetic characters.) Figure 12 shows that distributions are similar between computations using the stacking method and the ground-truth topology. In each case, the maximum-likelihood site corresponds to the true site. In the future, this test can be run on larger datasets involving more species, and more formal measures such as KL-Divergence can be used to compare distributions. The time it takes to run using the above data (from Stan inference to data analysis) on a standard personal laptop is a few minutes.

Data comes from genome.ucsc.edu.

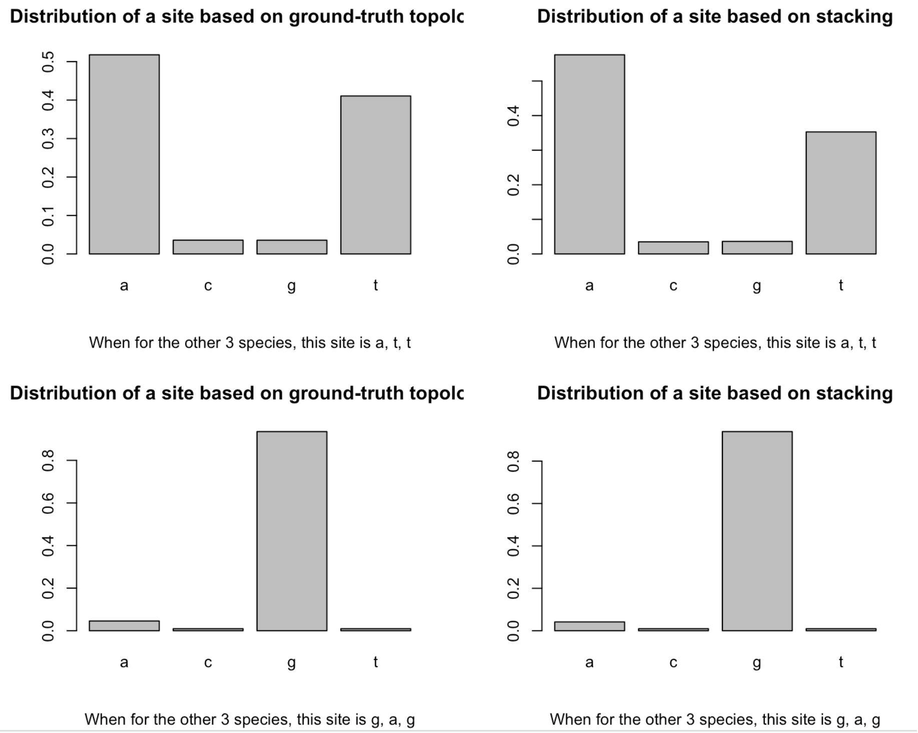


Figure 12: Posterior predicative distribution of two sites for Homo species. Top left and bottom left are distributions computed using ground-truth topology for each of the two sites. Top right and bottom right are distributions computed using stacking for each of the two sites.