**CoME 2019 Lecture 7 Notes**

**Model Specification**

Model-based inference—as the name implies—is all about the model. The model describes the process that gave rise to our data; in our case, the sequence data we observe in our study species was generated by a process of molecular evolution over a phylogenetic tree with branch lengths. Accordingly, if the model provides a ‘poor fit’ to the data, all bets are off.   
The model must balance two competing criteria: (1) the model must include adequate parameters to describe relevant aspects of the processes that gave rise to the sequence data, and; (2) the model must avoid superfluous parameters that only capture stochastic fluctuations in the data.

Failure to satisfy criterion (1) will result in *estimation bias*, such as systematic underestimation of the rate of substitution/branch lengths, inflated estimates of nodal support, phylogenetic errors associated with long-branch attraction, *etc*. To model the process accurately, the model must include parameters that capture relevant aspects of the process that gave rise to our data (unequal stationary frequencies, the degree of among-site rate variation, *etc*.). On the other hand, failure to satisfy criterion (2) will inflate *error variance* in parameter estimates, such as substitution rates, branch lengths, *etc*. Essentially, our dataset contains a finite amount of information to estimate parameters, so the error variance in the parameter estimates will increase as the number of parameters increase…we don’t want to waste this precious information on irrelevant parameters.

*Model Specification Issues*

Specifying a model requires consideration of three main issues; both maximum-likelihood and Bayesian approaches have been developed to address each of these three model-specification issues:

I. Model selection: *“What is the relative fit of two or more candidate models to my dataset?”*

1. Maximum-likelihood model-selection methods

(i) Hierarchical likelihood-ratio test (hLRT)

(ii) Akaike Information Criterion (AIC)

(iii) Bayesian Information Criterion (BIC)

2. Bayesian model-selection methods

(i) Bayes factors (BF)

II. Model adequacy: *“What is the absolute of a candidate model to my dataset?”*

1. Maximum-likelihood model-adequacy methods

Monte Carlo Simulation (parametric bootstrap)

2. Bayesian model-adequacy methods

Posterior-predictive simulation

III. Model averaging: *“How do we deal with uncertainty in the choice of model?”*

1. Maximum-likelihood model-averaging methods

Averaging by AIC model weights

1. Bayesian model-averaging methods

Reversible-Jump MCMC (rjMCMC)

**I. Model selection**

There are three maximum-likelihood-based model selection approaches (discussed below) and one Bayesian approach that are widely used:

*1. Hierarchical Likelihood Ratio Test (hLRT)*

We can assess the relative fit of two competing (and typically nested) models, *M0* and *M1*, to our data by comparing the ratio of their maximum-likelihood estimates. Formally, this involves calculation of a likelihood-ratio statistic for these data, which we write as follows:

Δ2**(ln***L*1 **ln***L*0**)**

(recall that the ratio of *X*/*Y* is equal to ln*Y* – ln*X*) where ln*L*1is the maximum-likelihood estimate (the probability of realizing the data under the model) under the more general model, *M*1, and ln*L*0is the maximum-likelihood estimate under the more restricted (‘null’) model, *M*1. Two models are *nested* if the simpler model is a restriction (*i.e.*, a simplification) of the other model. For example, JC69 is nested within the F81 model, which is obtained by constraining the stationary frequencies to be equal. The hLRT statistic, Δ, is approximately distributed as a Chi-square random variable, with degrees of freedom equal to the difference in the number of free parameters in the two models[[1]](#footnote-1): this is convenient in that it allows us to look up the significance of a given comparison in conventional statistical tables. Moreover, the hLRT provides a versatile framework for testing many hypotheses in a phylogenetic context (see especially the suggested reading: Huelsenbeck & Rannala, 1997).

In fact, the *correct* use of the hLRT in the context of models of nucleotide substitution entails testing prior hypotheses about the data (rather than model selection *per se*). For example, we may wish to ask whether our data exhibit a bias in the rates of transitions and transversions. This hypothesis could be evaluated by comparing a model that includes a parameter for this feature, *κ*, to a nested model that does not include a parameter to capture a bias in the transition/transversion substitution rates; *e.g.*, we might test this hypothesis by comparing the HKY85 to the F81 model. Note however, that we might have also chosen several other pairs of models to test this hypothesis (*e.g*., K80 and JC69).

There are several reasons that hLRT is not well suited for model selection (agnostically evaluating the fit of a dataset to a pool of candidate models). As just outlined, the sequence of parameter addition is often arbitrary (*e.g.*, whether we decide to test for the inclusion/exclusion of *κ* before or after including/excluding unequal stationary frequencies in the previous example). These hypothesis tests may have different outcomes in the context of other model parameters (such as *π* in our example). Moreover, the direction traveled through the graph of pairwise model comparisons is arbitrary (*i.e.*, whether we start at the most complex model and sequentially remove parameters, or start with the least complex model and add parameters). Again, this choice can change the optimal model identified by means of hLRT.

There are several other technical problems that arise when using hLRT for model selection: the ‘approximation’ of the test statistic to the Chi-square distribution is often quite poor for most datasets (especially near boundary conditions of the null model, such as if maximum-likelihood estimate of the base frequencies in F81 were close to 0.25), the approach is restricted to the comparison of nested models (at least using the Chi-square approximation, although Monte Carlo simulation of the null distribution is possible for non-nested models), and the approach is susceptible to multiple test problems; *i.e.*, if you perform many tests on a single dataset, you are increasingly likely to spuriously reject the simpler/null hypothesis (thus, inflating Type I error rates).

More fundamental problems associated with using the hLRT in the context of (substitution) model selection—at least in my opinion—involves three points:

(i)The approach assumes that maximum-likelihood estimates under the models are known without error, which is bogus (they are estimates from data, and are often associated with considerable uncertainty).

(ii)In order to retain the nesting of the two models under comparison, the tree is held constant over the two models, but if the model used can impact inferred tree, then there is no reason to believe that this convenience is justifiable.

(iii)hLRT is intended for *hypothesis testing* (assessing the relative support for an *a priori* prediction about the data), not for data exploration.

Surprisingly, hLRT remains one of the most commonly used method for substitution model selection in phylogenetics.

*2. Akaike Information Criterion* (*AIC*)

The Akaike information criterion (pronounced AKA-ee-kee) is an information theoretic metric that estimates the Kullback–Leibler (K–L) distance between the model under consideration and the true model that generated the data. The K–L distance attempts to capture the information lost in the data by using an incorrect model, so smaller values indicate better match of the candidate model to the data. Formally, this is simply calculating a score for each model as follows:

*AICi* 2**ln***Li* 2*pi*

where ln*Li* is the maximum-likelihood estimate of the data under model *i* that has *pi* free parameters. The AIC attempts to balance model fit (the first term, which is the MLE under the estimation model) and error variance (the second term, which is a penalty function based on the number of free parameters in model). The AIC scores can be individually computed for each candidate model and then we can compare the AIC scores for two compering models as follows:

Δ*AICi* *AICi* – **min***AIC*

which is simply the difference in the AIC scores of candidate model *i* and the model with the best AIC score. A more complex model is judged to significantly improve the fit to the dataset if the Δ*AIC* score exceeds some pre-specified threshold, Δ*AICcrit*. Relative to the hLRT, the AIC offers two main advantages for model selection: it allows more convenient comparison of non-nested models, and avoids multiple-test issues. However, like the hLRT, the AIC assumes ML estimates are known without error, and is a large sample approximation (*i.e.*, it is only theoretically correct when you have a very large sample of data). More troubling, the specification of the Δ*AICcrit* threshold is essentially arbitrary, and different thresholds can lead to the preference for different models, which underlies the relatively tenuous basis of the AIC on information theory (at least compared to the hLRT, which is based on the Central Limit Theorem).

*3. Bayesian Information Criterion* (*BIC*)

Don’t let the name fool you, the BIC is a maximum likelihood-based model selection method (sneaky!). This metric attempts to estimate the predictive ability of the model; specifically, the ability of the model to predict the data at hand. This is a Bayesian concept associated with the marginal likelihood of the data, but we’ll come back to that concept when we discuss Bayes factors, below. Formally, this simply entails calculating a score for each model as follows:

*BICi* 2**ln***Li* *pi* **ln***ni*

where ln*Li* is the maximum-likelihood estimate of the data under model *i* with *pi* free parameters, and *ni* is the number of independent observations (typically assumed to be the number of sites in the alignment). Like the AIC method, the BIC attempts to balance model fit (the first term, which is the MLE under the candidate model) and error variance (the second term, which is a penalty function based on the number of free parameters in model and the amount of information).

The BIC shares many of the advantages and limitations of the AIC (including the assumption that the maximum-likelihood estimates are known without error: how un-Bayesian is that!!), but is probably less biased toward more parameter-rich models than either AIC or hLRT (at least when the observations are at least moderate in number and truly independent). Like the AIC, the theoretical foundation for the BIC is less well established compared to other methods (hLRT and Bayes factors). In general, the bias of the three likelihood-based model selection methods toward more parameter-rich models is as follows:

*hLRT* Δ*AIC* Δ*BIC*

Accordingly, it’s important to keep in mind that the choice of different model-selection approaches may

impact the ‘optimal’ model that is selected.

*4. Bayes Factors*

The preceding model-selection methods are based on comparisons of the maximum likelihood of the data under the candidate models (which is why they are maximum-likelihood model selection methods). The Bayesian approach for selecting among candidate models/ hypotheses is based on comparing the *marginal likelihoods* of the candidate models: this is the *average* likelihood of the data, where the likelihood of the data is averaged over the priors (so it is a weighted average). The relative evidential support in the data for two alternative models (or hypotheses) can be evaluated using Bayes Factors. This approach compares the marginal likelihoods of the competing models/hypotheses. Recall that the marginal likelihood is the denominator of Bayes Theorem. With the marginal likelihood for the candidate models/hypotheses in hand, the Bayes factor is simply calculated as follows:



where *f*(**X**|*Mi*) is the marginal likelihood of the data, **X**, under model *i*. The interpretation of the Bayes factor is similarly straightforward: BF > 1 supports *M*1, whereas BF < 1 supports *M*2. This approach is both versatile (the models/hypotheses need not be nested), and is also relatively robust (as the marginal likelihood inherently accommodates uncertainty in model parameter values; it is the likelihood of the data integrated/averaged over the joint prior for all model parameters). More complex models are penalized by virtue of having to average the likelihood over a more high-dimensional joint prior.

Although the BF is *far* superior to the hLRT, AIC, *etc*. (it is more versatile and robust), it’s not a panacea. The tricky business is obtaining a reliable estimate of the marginal likelihoods of the candidate models. Recall that the magic of the Metropolis-Hastings algorithm (that is the foundation of most MCMC samplers) is that it avoids calculation of marginal likelihood (*i.e.*, these algorithms sample from the joint posterior by comparing the relative probabilities of sequential states). Therefore, we need to estimate the marginal likelihood somehow. Unfortunately, this is usually accomplished using the harmonic mean estimator, which is computationally efficient, but *extremely* unreliable. There are more reliable means of estimating the marginal likelihood, such as thermodynamic integration and stepping-stone methods. Not surprisingly, the tradeoff for the increased reliability of these estimators is an increase in the associated computational burden.

**II. Model Adequacy**

Even if we successfully identify the very best model from a pool of candidate models, the chosen model may nevertheless be woefully inadequate in an *absolute* sense. Accordingly, it is important to explore the adequacy of the fit of your model to the data. There are two very similar approaches, depending on whether you choose to proceed in an ML or Bayesian framework.

*1. ML Assessment of Model Adequacy: Monte Carlo Simulation (Parametric bootstrapping)*

Parametric bootstrapping is based on a simple premise: if the model under consideration provides an adequate description of the process that gave rise to the data at hand, then the model should also be able to predict ‘future’ datasets that are in some sense ‘similar’ to the data at hand. This involves a five-step process:

Step 1: Calculate a statistic from the original sequence data, *Tobs*. A statistic commonly used for this purpose is the multinomial likelihood, *T*, which captures aspects of the unique site patterns in the data (*e.g.*, column 1 of the data matrix may contain the site pattern AAGCCCC for our seven study species, and so on). The multinomial likelihood statistic is calculated as follows:



where **X**is our data matrix with *N* sites, *n* is the number of unique site patterns in the data matrix, *Θ*(*i*) is the *i*th unique site pattern, and *NΘ*(*i*)is the number of instances of *Θ*(*i*)in the data set. The specific statistic chosen is not terribly important, but the general idea is to summarize—in a single number—the pattern of variation in the data matrix (i.e., here is a number that indicates what our dataset looks like).

Step 2: Estimate the maximum-likelihood values for all parameters of the model from original data.

Step 3: Simulate replicate datasets (equal in size to the original dataset) using the inference model with the values of all parameters set to the maximum-likelihood values estimated in Step 2.

Step 4: Calculate the values of the statistic, *T*, for each simulated sequence dataset.

Repeat Steps 4–5 many times: the resulting test statistics generated from the simulated datasets collectively comprise a null distribution of the test statistic. This process should be familiar to you as good old Monte Carlo simulation.

Step 5: Compare the observed value to the resulting null distribution.

If the model under consideration provides an adequate description of the process that gave rise to the original dataset, then the test statistic for the observed dataset will fall near the center of the null distribution, otherwise, the statistic from the observed data will fall near the tail of the null distribution, indicating that the model cannot be used to predict future data that look like the observed dataset. We can compute the *p*-value by summing up all of the simulated values that are equal to or greater than the observed value of the test statistic, and diving that sum by the number of simulated replicates; a model that does a good job of predicting datasets that look like our observed dataset will not fall near either tail.

*2. Bayesian Assessment of Model Adequacy:* *Posterior Predictive Simulation*

If you understand the ML approach above, you are most of the way there to understanding the Bayesian equivalent; Posterior Predictive Simulation. The Bayesian approach for assessing the absolute fit of a candidate model to our dataset is very similar to the parametric bootstrap/Monte Carlo simulation approach used in maximum-likelihood estimation described above. The primary difference (and major advantage) is that the predictive distribution of the test statistic is generated by repeatedly drawing a set of parameter values from their respective joint posterior probability distributions to parameterize the generating model (rather than using the maximum-likelihood estimates as in parametric bootstrap/Monte Carlo simulation approach). Consequently, the predictive distribution accommodates uncertainty associated with the parameter estimates.

Step 1: Calculate a statistic from the original sequence data. A statistic commonly used for this purpose is the multinomial likelihood, *T*, which captures aspects of the unique site patterns in the data (*e.g.*, column 1 of the data matrix may contain the site pattern AAGCCCC, and so on). The multinomial likelihood statistic is calculated as follows:



where *X* is our data matrix with *N* sites, *n* is the number of unique site patterns in the data matrix, *Θ*(*i*) is the *i*th unique site pattern, and *NΘ*(*i*)is the number of instances of *Θ*(*i*)in the data set. The specific statistic chosen is not terribly important, but the general idea is to summarize—in a single number—the pattern of variation in the data matrix.

Step 2: Estimate the joint posterior probability density for all parameters of the model from original data using MCMC.

Step 3: Sample a set of parameter values from their corresponding joint posterior probability distribution estimated in step 2. Simulate a replicate dataset (equal in size to the original dataset) using the sampled parameter values.

Step 4: Calculate the value of the statistic, *T*, for each simulated sequence dataset.

Step 5: Repeat Steps 3–4 many times: the resulting test statistics generated from the simulated datasets collectively comprise a distribution of the test statistic predicted from the posterior distribution of parameters under the candidate model. Note that this process is called posterior predictive simulation.

Step 6: Compare the observed value to the resulting predictive distribution. Sum all predicted values of the test statistic greater or equal to the observed value and divide by the number of replicates that you performed. This quotient is called the posterior predictive *p*-value.

If the model under consideration provides an adequate description of the original data, then the test statistic for the observed data will fall near the center of the null distribution, indicating that data can be generated under the model that look like the original data. Otherwise, the statistic from the observed data will fall near the tail of the null distribution, indicating that the model is inadequate. We can compute the posterior-predictive *p*-value in the same way as we compute the *p*-value for parametric bootstrapping. Although this is generally used to assess the absolute fit of our dataset to a given model, it can also be used to assess the fit of multiple candidate models. In this case, we would be comparing the candidate model by virtue of their absolute fit to our dataset.

**III. Model Uncertainty**

*1. ML Approach: Model Averaging Based on AIC Weights*

If the fit of the data to the best model is substantially better than the fit to the other candidate models, conditioning on the best model may be justifiable. Conversely, if the fit of the data to the best model is only marginally better than the fit to some set of other candidate models, model averaging may be necessary. The maximum-likelihood solution to this problem is to first calculate the AIC weight for each of the candidate models (see above). We then identify the 95% confidence set of models by iteratively summing the AIC model weights starting from the model with the highest weight until the cumulative weight of the set of models is ≥ 95%. We then perform a series of independent ML estimates for each of the models in the confidence set, such that we have maximum likelihood estimates (MLEs) for each parameter for each of the models in the set. Finally, to accommodate for model uncertainty, we calculate the MLE for each parameter as the weighted average over the confidence set of models, where the weighting is based on the AIC weights of each model. If you are saying to yourself “Boy, that sounds like a kludge…surely there must be a better way!”, you are correct (see below).

*2. Bayesian Approach: Model Averaging by Reversible Jump MCMC*

As mentioned above, if the fit of our dataset to the best model is substantially better than the fit to the other models, conditioning on the best model (*i.e*., making it a fixed assumption of the analysis) may be justifiable. Conversely, if the fit of the data to the best model is only marginally better than the fit to the other models, model averaging may be necessary. The Bayesian perspective on this problem is “*If there is uncertainty in model specification, let’s accommodate it!*” Specifically, we simply treat the model as a random variable and average inferences over the space of candidate models. The idea here is to treat the MODEL as a random variable, and average estimates over the expanse of model space, while simultaneously averaging over prior probability densities for each parameter of each model.

Because different models may differ in the number of free parameters, this entails changes in the dimensionality of parameter space, which is accommodated by using a trans-dimensional MCMC algorithm that jumps between dimensions in model space (specifically, this involves the third term in the equation used to compute acceptance probabilities, called the proposal ratio or Hastings ratio). The proportion of time that the RJ MCMC visits a given model is an approximation of the marginal posterior probability of that model.

1. A *free parameter* is one that is estimated from the data—it is ‘free to vary’, rather than fixed to some value *a priori*. [↑](#footnote-ref-1)