



Automating Bayesian Analysis of Partitioned Sequence Datasets

JOHN P. HUELSENBECK¹ AND BRIAN R. MOORE²

¹*Department of Integrative Biology, University of California, Berkeley
3140 VLSB, Berkeley, CA 94720-3140, U.S.A.*

²*Department of Evolution and Ecology, University of California, Davis
Storer Hall, One Shields Avenue, Davis, CA 95616, U.S.A.*

Background

Our motivation for developing **AutoParts** is to estimate phylogeny under a model that accommodates process heterogeneity—variation in the evolutionary process across the sequence alignment—within a flexible Bayesian statistical framework.

We define a *partition scheme* as the number of distinct substitution processes and the assignment of (subsets of) nucleotides sites to those distinct processes. Our solution for accommodating process heterogeneity treats the partition scheme as a random variable described by the Dirichlet process prior (DPP) model. Numerical integration (MCMC simulation) is then used to estimate parameters of the phylogenetic model—the tree topology, branch lengths and substitution model parameters—while averaging over all possible partition schemes.

The user is required to specify the sequence alignment, the data partitions (the subsets of nucleotide sites intended to capture process heterogeneity), and the hyper parameters of the DPP model. Parameters of the DPP model include (1) the ‘concentration’ parameter, α , which controls the expected number of partition schemes, and (2) the ‘base measure’, G_0 , which specifies the prior probability density for each substitution model parameter.

AutoParts implements a Dirichlet process for each of the four substitution model parameters: the stationary frequencies, the exchangeability rates, the tree length, and the alpha parameter controlling the degree of among-site rate variation. The Dirichlet processes for the four substitution model parameters share a common concentration parameter, which can be specified either as a fixed value or treated as a random variable (described by a second-order gamma hyperprior).

In this guide, we briefly describe how to install **AutoParts**, and then walk through a simple analysis of the included conifer dataset, illustrating how to generate and interpret results.

Details of the methods implemented in **AutoParts** are described in the following paper:

Moore, B. R., J. McGuire, F. Ronquist, and J. P. Huelsenbeck. (in review) Bayesian analysis of partitioned data. *Systematic Biology*.

Details of the **AutoParts** software are described in the following paper:

Moore, B. R., S. Höhna, and J. P. Huelsenbeck. (in review) Automating the Bayesian analysis of partitioned sequence datasets with **AutoParts**. *Bioinformatics*.

Installing AutoParts

Download the **AutoParts** archive, decompress it, and move it to the desired location on your computer. If you are using a Mac OS, you may be able to use the pre-compiled binary included with the **AutoParts** distribution. To execute the binary, navigate to the **bin** directory within the **AutoParts** bundle:

```
cd users/<path to AutoParts directory on your computer>AutoParts/bin/ <return>
```

Next, execute the program by typing: `./AutoParts <return>`

If the precompiled binary is not compatible with your OS, you can compile it from the included source code using the **cmake** utility. Point your terminal to the directory that contains the **AutoParts** source

code, which is located in the program bundle:

```
cd users/<path to AutoParts directory on your computer>AutoParts/src/ <return>
```

Next, compile the program by first typing:

```
cmake . <return>
```

followed by:

```
make <return>
```

If you do not have **cmake** installed, you can install it using the **homebrew** package-manager utility by typing:

```
brew install cmake <return>
```

If you do not have the **homebrew** package-manager utility, you can install it by first navigating to your home directory by typing:

```
cd <return>
```

and then typing the following in your terminal window:

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/homebrew/go/install)" <return>
```

If you want to be able to execute **AutoParts** from any directory on your computer, you can add it to your path environment variable. On a Mac, this can be achieved by editing the **.bash_profile** file. On my laptop, the **AutoParts** binary is located in the path **usr/Applications/AutoParts/src/**; I will add this path to my bash profile using the **vi** text editor. Type:

```
vi .bash_profile <return>
```

The **vi** editor will display something like this:

```
# Setting PATH for Python 2.7
# The original version is saved in .bash_profile.pysave
PATH="/Library/Frameworks/Python.framework/Versions/2.7/bin:${PATH}"
export PATH
export PATH="/usr/Applications/mrbayes_3.2.2/src:${PATH}"
export PATH="/usr/Applications/RevBayes/src:${PATH}"
~
~
~
".bash_profile" 8L, 423C
```

Type **i** to switch to **INSERT** mode, enter the path to the **AutoParts** binary, press the **<Esc>** key to go back to **command** mode, and then type **:wq** to save and quit. Finally, type **source ~/.bash_profile <return>** to reload the Terminal so it will read in the new path.

Now you can execute **AutoParts** from any directory on your computer by typing **AutoParts <return>**.

Input file format

AutoParts uses a simple Phylip-style input file format. This is a plain text file that includes three main pieces of information:

1. The number of species followed by the number of sites (specified in first line of the file)
2. The multiple sequence alignment
3. The set of two or more predefined data partitions (specified as charsets at the end of the file)

For example, here is part of the input file for the included example file, **conifer.in** (showing the first 12 sites of the sequence alignment):

Box 1: A fragment of the input file containing the 9 conifer species in the included example dataset.

```
9 2659
Ginkgo_biloba      TTATTGGTCCAG...
Araucaria_araucana -----GGTCCGG...
Cedrus_deodara     TCATTGGCCCAG...
Cupressus_arizonica -----...
Juniperus_communis -----...
Pinus_densiflora   TCATTGGCCCAG...
Podocarpus_chinensis TCATCGGCCCTG...
Sciadopitys_verticillata TCATTGGTCCAG...
Taxus_baccata      TTATCGGCCCCAG...
charset atpB_1st = 1-1394\3;
charset atpB_2nd = 2-1394\3;
charset atpB_3rd = 3-1394\3;
charset rbcL_1st = 1395-2659\3;
charset rbcL_2nd = 1396-2659\3;
charset rbcL_3rd = 1397-2659\3;
```

Our example file contains an alignment for 9 conifer species with 2659 sites sampled from two chloroplast protein-coding genes, *atpB* and *rbcL*. It is up to the user to specify a set of data partitions that are likely to capture patterns of process heterogeneity across the alignment. In this case, for example, we might specify two data partitions, one for the *atpB* gene and the second for the *rbcL* gene. Alternatively, we might specify three data partitions corresponding to the three codon positions of these protein-coding genes. In any case, the set of data partitions that we specify becomes a *fixed assumption of the analysis*.

Note, however, that it is possible to explore alternative data-partitioning schemes—performing a separate analysis under each candidate scheme—and then choosing among them based on the marginal likelihoods of competing schemes. Generally, we recommend erring on the side of defining more granular data partitions: if the evolutionary process is uniform within two pre-specified data partitions, they will be grouped together under the DPP model. Here, we have decided to define separate partitions for the three codon positions in each of the two genes, for a total of $K = 6$ data partitions.

Specifying details of the analysis

In this guide, we will use **AutoParts** interactively by typing commands in the command-line console. To see a list of available commands, you can execute **AutoParts** and view the splash page:

Box 2: The **AutoParts** splash page listing available commands.

```
Usage:
-i : Input file name
-t : Tree file name (for constraining the analysis to a fixed tree)
-o : Output file name
-l : Number of MCMC cycles
-d : Number of MCMC cycles to discard as the "burn-in"
-p : Print frequency
-s : Sample frequency
-b : Exponential parameter for branch lengths
-g : Number of discrete gamma rate categories
-e : Exponential parameter for shape parameter describing ASRV
-c : Concentration parameter is fixed (1) or a random variable (0)
-k : Prior mean of the number of categories when the concentration parameter is fixed
-m : Prior mean of the concentration parameter when it is a random variable
-v : Prior variance of the concentration parameter when it is a random variable
-t1 : MCMC tuning parameter for the tree topology parameter
-t2 : MCMC tuning parameter for the gamma shape parameter
-t3 : MCMC tuning parameter for the base frequencies
-t4 : MCMC tuning parameter for the substitution rates
-t5 : MCMC tuning parameter for the tree length parameter

Example:
./AutoParts -i <input file> -o <output file>
```

We will specify details for an analysis of the conifer dataset, where we fix the concentration parameter (using the **-c 1** argument) such that the prior mean on the expected number of process patrons, $E(k)$, is centered on an intermediate value for the $K = 6$ data partitions (using the **-k 3.0** argument). We will run the chain for 100,000 cycles (using the **-l 100000** argument) and thinned by sampling every 100 cycles (using the **-s 100** argument), and we will have it print to the screen at the same frequency (using the **-p 100** argument). It will generally be necessary to carefully diagnose the MCMC output to assess whether the chain has converged to the stationary distribution, and if so, to identify the fraction of the pre-stationary samples that should be discarded as burn-in. For the moment, however, we will boldly pre-specify the burn-in as the first half of the samples (using the **-d 50000** argument).

The details of this analysis are specified by the following command-line string:

```
AutoParts -i conifer.in -o conifer -l 100000 -p 100 -s 100 -d 50000 -c 1 -k 3.0 <return>
```

Note that the screen output is automatically written to a file called **<output_file_name>.log**, which contains important summary information and diagnostics that we will discuss below. For this reason, it's generally a good idea to match the sampling and printing frequencies (using the **-s** and **-p** arguments).

Running the analysis

Hit **<return>** to start the MCMC simulation (the analysis required ~ 5 minutes to complete on my laptop). When the analysis begins, it reports the state sets—the number of process partitions for each of the four parameters, the assignment of data partitions to those process partitions, and the parameter values for the process partitions:

Box 3: The start of the **AutoParts** screen output.

```
AutoParts v.1.0
John Huelsenbeck & Brian Moore
University of California, Bervis

Setting up state sets
alpha fixed to 1.69577
Table 1
Patrons at table: "012345"
 0 (9, -1, -1) 0.00000 (Ginkgo_biloba)  <- Root
 9 (10, 14, 0) 0.00897
10 (1, 3, 9) 0.04196
 1 (-1, -1, 10) 0.03625 (Araucaria_araucana)
 3 (-1, -1, 10) 0.09207 (Cupressus_arizonica)
14 (15, 11, 9) 0.25955
15 (7, 8, 14) 0.02357
 7 (-1, -1, 15) 0.01012 (Sciadopitys_verticillata)
 8 (-1, -1, 15) 0.00880 (Taxus_baccata)
11 (12, 2, 14) 0.04422
12 (13, 5, 11) 0.00523
13 (4, 6, 12) 0.03358
 4 (-1, -1, 13) 0.11893 (Juniperus_communis)
 6 (-1, -1, 13) 0.20797 (Podocarpus_chinensis)
 5 (-1, -1, 12) 0.00497 (Pinus_densiflora)
 2 (-1, -1, 11) 0.10382 (Cedrus_deodara)
alpha fixed to 1.69577
Table 1
Patrons at table: "04"
alpha = 0.19929
Table 2
Patrons at table: "1"
alpha = 0.29665
Table 3
Patrons at table: "235"
alpha = 1.95620
```

The output first indicates the value (and treatment) of the concentration parameter, which for this analysis is **alpha fixed to 1.69577**: this value centers the prior for the number of process partitions to the specified value, $E(k) = 3.0$. Note that α is conventionally used for the concentration parameter of the DPP model, as it is in **AutoParts**. In our paper (Moore et al., in review), however, we used χ for this parameter. Our motivation was to avoid confusion between the concentration parameter of the DPP model and the familiar α -shape parameter of the discrete-gamma model used to accommodate among-site rate variation (ASRV).

The output specifies the initial state of the chain with respect to the process partitions for each parameter in the following order: tree topology, alpha-shape parameter, the exchangeability/substitution rates, base frequencies, and tree length. By assumption, all $K = 6$ data partitions share the same process partition for the tree parameter. The ‘value’ of the tree parameter (the topology) has been drawn from its ‘base measure’—the uniform prior distribution on tree topologies—and is depicted in tab-delimited format.

Next, the output describes the initial process partition for the alpha-shape parameter. Three process partitions have been sampled. Data partitions 0 and 4 (corresponding to the first codon positions of *atpB* and *rbcL*, see Box 1) have been assigned to the first process partition, which has an alpha value of 0.19929. Data partition 1 (corresponding to the second codon position of *atpB*) has been assigned to the second process partition with a parameter value of 0.29665. Finally, the remaining data partitions, 2, 3, 5 (corresponding to the three codon positions of *rbcL*, have been assigned to the third process partition with a parameter value of 1.95620. Again, this initial partition scheme for the α -shape parameter is based on a random draw from the DPP model. The concentration parameter controls the number of process partitions and the assignment of data partitions to tables. The base measure, G_0 , is the prior from which parameter values are drawn (we use an exponential prior for the α -shape parameter controlling the degree of ASRV).

The output continues in this way, describing the full model specification that corresponds to the initial (and random) state of the chain. Samples are then written to the screen at the specified interval (here every 100 cycles):

Box 3: The first 15 samples from the MCMC simulation.

```

Gen -- lnL          -- Number of occupied tables  -- Parameter update
100 -- -9949.634 -- t(1) rv(3) sr(3) bf(2) tl(4) -- updating tree
200 -- -9767.748 -- t(1) rv(3) sr(2) bf(3) tl(4) -- updating tree
300 -- -9684.823 -- t(1) rv(3) sr(2) bf(2) tl(3) -- updating substrates
400 -- -9685.444 -- t(1) rv(3) sr(2) bf(2) tl(3) -- updating tree
500 -- -9646.255 -- t(1) rv(3) sr(2) bf(2) tl(3) -- updating tree
600 -- -9629.956 -- t(1) rv(3) sr(2) bf(2) tl(3) -- updating tree
700 -- -9603.426 -- t(1) rv(3) sr(2) bf(2) tl(4) -- updating tree
800 -- -9596.978 -- t(1) rv(3) sr(3) bf(2) tl(4) -- updating tree
900 -- -9600.097 -- t(1) rv(3) sr(2) bf(2) tl(3) -- updating tree
1000 -- -9613.031 -- t(1) rv(4) sr(3) bf(2) tl(3) -- updating tree
1100 -- -9547.029 -- t(1) rv(2) sr(3) bf(2) tl(5) -- updating basefreq
1200 -- -9587.855 -- t(1) rv(2) sr(2) bf(2) tl(4) -- updating substrates
1300 -- -9529.674 -- t(1) rv(3) sr(3) bf(2) tl(4) -- updating tree
1400 -- -9482.361 -- t(1) rv(3) sr(2) bf(2) tl(4) -- updating tree
1500 -- -9462.873 -- t(1) rv(3) sr(3) bf(2) tl(4) -- updating tree

```

For each sample, the screen logs the generation of the MCMC, the log likelihood (*lnL*), the number of process partitions for each of the five parameters, and the parameter that is currently being operated on. For example, the tenth sample (generation 1000), there is one process partition for the tree topology, **t(1)**, the alpha-shape parameter has four process partitions, **rv(4)**, there are three process partitions for the substitution rates, **sr(3)**, there are two process partitions for the base frequencies, **bf(2)**, and one process partition for the tree length, **tl(1)**. At that cycle, a change to the tree topology has been proposed.

When the MCMC simulation is complete, several output files will be generated:

1. Sampled partitions for each model parameter: **<input_file_name>.<parameter>.part**
2. Sampled values for each model parameter: **<input_file_name>.<parameter>.out**
3. A majority-rule consensus of sampled trees: **<input_file_name>mrc.tre**
4. A credible set of sampled trees: **<input_file_name>CI.trees**
5. Sampled log-likelihood values: **<input_file_name>.lnL**
6. A log of the screen output: **<input_file_name>.log**

Interpreting the results

When the MCMC analysis is complete, **AutoParts** automatically computes some useful values that are written to the `.log` file. Specifically, it reports the acceptance rates to help assess mixing of the MCMC simulation: as a rule of thumb, acceptance rates should fall within the 20% – 70% range. It also calculates the various estimates of the marginal likelihood, including the AICM (Raftery et al., 2007), the original (Newton and Raftery, 1994) and modified (Suchard, Weiss and Sinsheimer, 2001) harmonic-mean estimator, and the smoothed marginal-likelihood estimator. In principle, these marginal likelihoods could be used to evaluate various hypotheses using Bayes factors (e.g., Suchard, Weiss and Sinsheimer, 2001). We caution, however, that these marginal-likelihood estimates are extremely crude and should only be used as a very rough guide. Rigorous hypothesis tests will require more stable marginal-likelihood estimates, which could be estimated using thermodynamic integration (e.g., Lartillot and Philippe, 2006) and/or stepping-stone simulation (e.g., Fan et al., 2011; Xie et al., 2011).

Box 4: Some summary statistics computed at the end of the MCMC simulation.

```
Acceptance Rates:
Parameter    Tries  Accep  Rate
tree         42787  4985   0.12
asrv         14149  12947  0.92
subrates     14377  10790  0.75
basefreq     14304  8326   0.58
treelength   14383  12144  0.84

Marginal lnL Estimates:
AICM                      = -19273.99
Harmonic mean (old way)   = -9422.36
Harmonic mean (Suchard)   = -9422.36
Marginal likelihood (smoothed) = -9383.81
```

Next, **AutoParts** summarizes information regarding inferred patterns of process heterogeneity. This information is reported for each of the model parameters in the following order: the topology (**tree**), alpha-shape parameter the topology (**asrv**), exchangeability parameters (**subrates**), stationary frequencies (**basefreq**), and tree length (**treelength**).

For each parameter, **AutoParts** reports some useful and interesting summaries:

1. The mean partition scheme (the ‘average partition’)
2. The marginal and cumulative probability of sampled process partitions
3. The posterior and prior probability for the number of process partitions
4. The probability that any pair of data partitions belong to the same process partition

Let’s explore these summaries for the α -shape parameter, depicted below (Box 5):

Box 5: Summary of the inferred pattern of process heterogeneity for the alpha-shape parameter.

```

Restaurant: asrv
Average Partition = (1,2,1,3,3,3) [6,3]
Unique partitions:
  1 -- 41941 0.8388 0.8388 (1,2,1,3,3,3) [6,3]
  2 -- 7536 0.1507 0.9895 (1,2,3,4,4,4) [6,4]
  3 -- 362 0.0072 0.9968 (1,2,1,3,4,3) [6,4]
  4 -- 82 0.0016 0.9984 (1,2,3,4,5,4) [6,5]
  5 -- 42 0.0008 0.9992 (1,2,1,3,4,4) [6,4]
  6 -- 35 0.0007 0.9999 (1,2,1,2,2,2) [6,2]
  7 -- 3 0.0001 1.0000 (1,2,1,3,2,3) [6,3]
Posterior and prior probability distributions for the number of tables:
  1 -- 0 0.0000 0.0692
  2 -- 35 0.0007 0.2560
  3 -- 41944 0.8389 0.3659
  4 -- 7940 0.1588 0.2340
  5 -- 82 0.0016 0.0670
  6 -- 0 0.0000 0.0079
Probability that patrons are seated at the same table:
  1 2 -- 0.0000
  1 3 -- 0.8476
  1 4 -- 0.0000
  1 5 -- 0.0000
  1 6 -- 0.0000
  2 3 -- 0.0000
  2 4 -- 0.0007
  2 5 -- 0.0008
  2 6 -- 0.0007
  3 4 -- 0.0000
  3 5 -- 0.0000
  3 6 -- 0.0000
  4 5 -- 0.9902
  4 6 -- 0.9992
  5 6 -- 0.9911

```

1. Process partitions

First, **AutoParts** reports the mean partition scheme for the α -shape parameter: this is the partition scheme that minimizes the squared distance to all other sampled partition schemes for this parameter (Gusfield, 2002). The mean partition scheme is rendered in ‘restricted-growth form’ (RGF) notation (Stanton and White, 1986). Here, the mean partition scheme for the α -shape parameter (in RGF notation) is (1, 2, 1, 3, 3, 3). This indicates that the six data partitions are distributed among three process partitions as follows: the first and third data partitions (the first and third codon-position sites of the *atpB* gene region) share a common degree of among-site rate variation, the degree of ASRV for the third data partition (the second codon position of the *atpB* gene region) is inferred to be unique, and data partitions 4, 5, and 6 (for the three codon-positions of the *rbcL* gene region) are inferred to share a common degree of ASRV.

Indeed, the process partition for the α -shape parameter inferred under the DPP model agrees well with the corresponding estimates of the marginal posterior probability densities for these data partitions, plotted below (Figure 1) by loading the corresponding log file (**conifer.asrv.out**) in **Tracer** (Rambaut and Drummond, 2007).

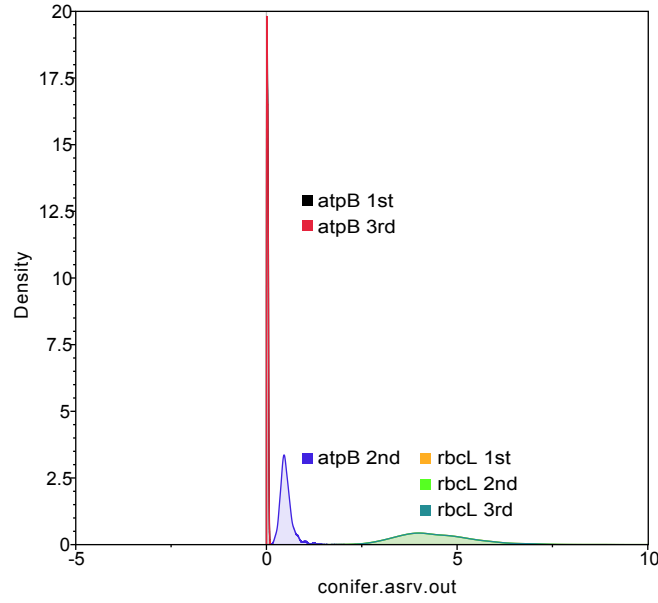


Figure 1: Inferred patterns of among-site substitution rate variation within the conifer sequence alignment. The marginal posterior probability densities are plotted from the corresponding log file for the α -shape parameter (`conifer.asrv.out`) estimated for each of the $K = 6$ predefined data partitions.

The process partitions for all four substitution-model parameters are summarized in Figure 2. We provide an R script (`AutoPlots.R`) that will generate this figure from the `conifer.log` file.

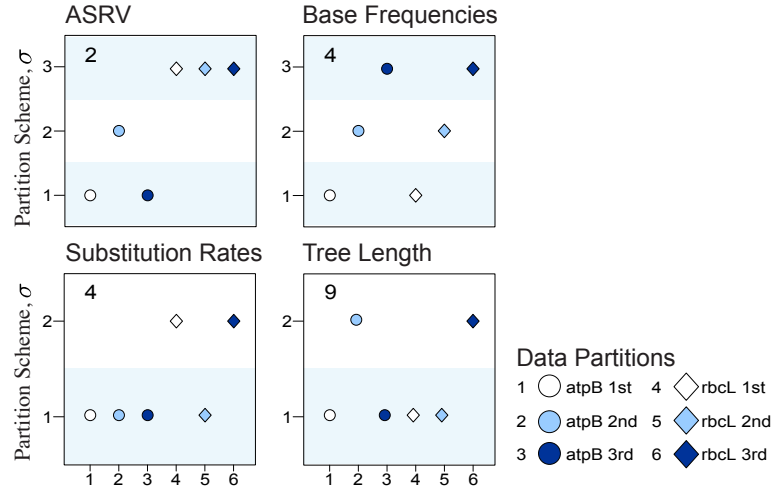


Figure 2: Summarizing process heterogeneity in conifers. For example, the $K = 6$ data partitions are allocated to $k = 3$ distinct process partitions for the α -shape parameter (the shaded rows of the panel), where the allocation vector is $\mathbf{z} = (1, 2, 1, 3, 3, 3)$ (depicted by the assignment of the data-partition symbols to their respective rows). The number of process partitions in the 95% credible set is depicted in the upper left of each panel.

To use `AutoPlots.R`, navigate to the `AutoPlots` directory within the `AutoParts` bundle:

```
cd users/<path to AutoParts directory on your computer>AutoParts/AutoPlots/ <return>
```

Now execute `AutoPlots.R` by typing: `Rscript AutoPlots.R --args <path_to_file><input_file_name>.slog`.

A figure named `<input_file_name>.pdf` will be written to the same directory as the `.slog` file.

2. Marginal and cumulative probability of unique process partitions

The **AutoParts** post-analysis report lists each unique process partition ordered by marginal posterior probability (Box 5). This facilitates identification of the 95% credible interval (CI) of process partitions, *i.e.*, by adding the marginal probability for the first process partition to that for the second process partition, and then adding this sum to the marginal probability for the third process partition, and so on, until the *cumulative* probability is equal to or greater than 0.95. The size of the 95% CI therefore indicates the degree of uncertainty in the estimated process partition for each parameter. The size of the credible set for each parameter is depicted in the upper-left corner of the corresponding panel in Figure 2.

3. Posterior and prior probability for the number of process partitions

The DPP model allows us to calculate the prior probability for the number of process partitions, k , given a specified value for the concentration parameter, α and the number of data partitions, K :

$$\mathbb{P}(k \mid \alpha, K) = \frac{\mathcal{S}_1(K, k) \alpha^k}{\prod_{i=1}^K (\alpha + i - 1)}, \quad (1)$$

where $\mathcal{S}_1(\cdot, \cdot)$ is the Stirling number of the first kind, which specifies the possible permutations of K data partitions among k process partitions:

$$\mathcal{S}_1(n, k) = (-1)^{n-k} \binom{n}{k}.$$

AutoParts reports the prior probabilities (computed analytically using the above equations) and the corresponding posterior probabilities (estimated numerically using MCMC) for the number of process partitions for each of the substitution model parameters. For example, the post-analysis summary reports the prior/posterior probabilities ($P = 0.3659, 0.8389$, respectively) of $k = 3$ process partitions for the α -shape parameter (Box 5). This provides an efficient and flexible framework for comparing competing models/testing alternative hypotheses regarding the nature of process heterogeneity in molecular evolution (discussed below).

4. Posterior probability that two data partitions share the same process partition

The DPP model also allows us to calculate the prior probability that any two data partitions, x_i, x_j belong to the same process partition, given a specified value for the concentration parameter, α :

$$\mathbb{P}(x_i = x_j \mid \alpha) = \frac{1}{1 + \alpha}. \quad (2)$$

As in the case for the number of process partitions, the prior probabilities that two data parts share the same process partition (which can be calculated analytically using the above equation) can be compared to the corresponding posterior probabilities (which are estimated numerically using MCMC). For example, the post-analysis report indicates that data partitions 1 and 3 (the first and third position sites of the *atpB* gene) share the same process partition for the α -shape parameter with probability $P = 0.8476$ (Box5).

Diagnosing MCMC performance and assessing prior sensitivity

Diagnosing MCMC performance.—Model-based inference is, after all, based on the model. Careful research means being vigilant regarding the choice of model and also rigorously assessing our ability to estimate under the chosen model. The first issue—model specification—is critically important for the simple reason that unbiased estimates can only be obtained under a model that provides a reasonable description of the process that gave rise to our data. The DPP model implemented in **AutoParts** is particularly attractive because it effectively obviates need to select among (mixed) models by virtue of averaging inference of phylogeny over all possible process partitions.

The second issue—rigorously assessing the ability to obtain reliable estimates under the chosen model(s)—typically receives less attention. The implicit assumption, it seems, is that if a model is implemented correctly, and if that implementation is used to obtain an estimate from a given dataset, then we must have performed valid inference under the model. This would be perfectly sound reasoning if inferences were based on analytical methods. Owing to the complexity of the models, however, it is not possible to estimate phylogenetic model parameters analytically. Instead, parameter estimates are based on numerical methods. In the Bayesian statistical framework, we use Markov chain Monte Carlo (MCMC) algorithms to approximate the joint posterior probability density of phylogenetic parameters. We may be comforted to know that, in theory, an *appropriately constructed* and *adequately run* MCMC simulation is guaranteed to provide an arbitrarily precise description of the posterior probability density (Tierney, 1994). In practice, however, even a given MCMC algorithm that provides reliable estimates in *most* cases will nevertheless fail in *some* cases and is not guaranteed to work for any particular dataset. This raises an obvious question: “When do we know that an MCMC algorithm provides reliable estimates for a given empirical analyses”. The answer is simple: *Never*.

Accordingly, the general dictum of Bayesian inference—vigilant diagnosis of MCMC performance—is particularly crucial for estimation of phylogeny (and other model parameters) under high-dimensional mixture models, such as the Dirichlet process prior approach implemented in **AutoParts**. We recommend carefully evaluating the output generated by **AutoParts** using diagnostic tools such as **bonsai** (May et. al. 2014). Poor mixing for a given parameter can be addressed by altering the value of the corresponding tuning parameter (Box 2). The tuning parameters control the magnitude of the proposed changes to current parameter values. If acceptance rates for a parameter are too low, the value of the corresponding tuning parameter should be decreased (and vice versa); this is achieved using the `-t<integer>` arguments (Box 2). Some aspects of the MCMC output under the DPP model warrant brief comment. For example, although multi-modal marginal posterior probability densities are generally cause for concern, under the DPP they may be innocuous. Because of the discrete nature of process partitions, the marginal posterior probability density for a parameter may exhibit more than one mode if there is uncertainty regarding the allocation of that data partition to alternative (discrete) process partitions.

In fact, the identification of a multi-modal marginal posterior probability densities may indicate that the pre-defined data partition includes residual process heterogeneity for that parameter. Imagine, for example, that we defined a data partition comprising all of the second-position sites of the *atpB* gene, and—after running the analysis—observed that the marginal posterior probability density for this data partition had two modes for the tree-length parameter: one at a lower substitution rate, and the other at a higher rate. This might be evidence of a pathological parameter interaction (*e.g.*, between the α -shape and tree-length parameters). On the other hand, the multiple modes may indicate that the data partition as specified contains residual variation in overall substitution rate; for example, some subset of these second position sites may be under positive selection. Careful MCMC diagnosis is required to validate the approximation of the joint posterior probability density, and may also provide opportunities to make novel insights into the processes that gave rise to your data.

Exploring the impact of the concentration parameter.—The Bayesian statistical framework views parameters as random variables, which requires that we specify probability distributions describing the nature of their random variation. Formally, a prior specifies our belief about the parameter values *before* evaluating the data at hand. The chosen prior is then updated by the information in the data via the likelihood function, transforming it into the corresponding posterior probability distribution (reflecting our beliefs about the parameter value *after* evaluating the data at hand). Because the posterior probability is proportional to the product of the likelihood and the prior, the chosen prior will always influence the posterior to *some* degree. This naturally raises concerns regarding the sensitivity of our estimates to the chosen prior. The DPP model implemented in **AutoParts** has two main priors: the concentration parameter, α —which controls the expected number of process partitions, $E(k)$ and the allocation of data partitions among those process partitions, \mathbf{z} —and the base measure, G_0 , which controls the prior values for each process partition. The base measures implemented in **AutoParts** are the generally familiar prior probability distributions for phylogenetic model parameters:

$$\begin{aligned}\tau &\sim \text{Discrete Uniform}(1, \dots, B(S)) \\ \mathbf{p} &\sim \text{Dirichlet}(1, 1, \dots, 1) \\ T &\sim \text{Gamma}(2S - 3, \lambda_1) \\ \mathbf{r} &\sim \text{Dirichlet}(1, 1, 1, 1, 1, 1) \\ \boldsymbol{\pi} &\sim \text{Dirichlet}(1, 1, 1, 1) \\ \alpha &\sim \text{Exponential}(\lambda_2)\end{aligned}$$

where $B(S) = (2S - 5)!!$ is the number of possible binary, unrooted trees for S species.

By contrast, the concentration parameter of the DPP model is relatively novel, and warrants more attention. Our experiments with empirical and simulated datasets indicate that estimates under the DPP model are sometimes sensitive to the specified concentration parameter. This is particularly troubling because we generally will not have access to a clear, biologically motivated prior on the number of process partitions. Accordingly, we strongly encourage you to assess the impact of the concentration parameter on estimates. Specifically, we recommend adopting one of two strategies:

Strategy 1. If you choose to treat the concentration parameter as a fixed value (using the `-c 1` argument, Box 2) we recommend exploring a range of α values that center the prior mean for the number of process partitions, $E(k)$, that span the entire range of possible values. For example, the conifer dataset includes $K = 6$ data partitions, which would minimally require three separate analyses with α iteratively specified at an intermediate and two extreme values; *e.g.*, $E(K) = \approx 1.0, 3.0, \approx 6.0$, using the `-k <real_number>` argument (Box 2). Note that the prior mean cannot be centered exactly on the boundaries values (*e.g.*, at 1.0 or 6.0); instead, values close to these boundaries can be used (*e.g.*, 1.2 or 5.8).

Strategy 2. If you choose to treat the concentration parameter as gamma-distributed random variable (using the `-c 0` argument, Box 2), we recommend using gamma hyperpriors that specify a diffuse prior for the number of process partitions, $E(k)$, that is centered on an intermediate value (using the `-m` and `-v` arguments to control the mean and variance of the gamma hyperprior on α , respectively; Box 2). We prefer this hierarchical Bayesian solution for three reasons: (1) It is consistent with the principles of Bayesian inference to use a (hyper) prior to describe our uncertainty regarding a parameter value; (2) Our experience with real and simulated data indicates that they typically contain sufficient information to allow the concentration parameter to be reliably estimated from the data under a hierarchical approach, and; (3) The hierarchical approach for specifying the concentration parameter effectively accommodates uncertainty in this parameter value while decreasing the number of required MCMC simulations.

Testing molecular evolutionary hypotheses with AutoParts

Our motivation for developing the DPP approach is to provide more robust estimates of phylogeny (tree topology and branch lengths) by virtue of integrating over all possible patterns of process heterogeneity across the alignment (where the process partitions are effectively nuisance parameters). However, an equally fruitful application of the DPP approach would focus directly on the nature of process heterogeneity to study various aspects of molecular evolution (where the phylogeny is essentially a nuisance parameter), providing a flexible framework to directly study how various aspects of the substitution process—stationary frequencies, exchangeability rates, overall substitution rate and the degree of ASRV—vary within and among gene/omic regions.

In a Bayesian framework, we compare two competing hypotheses/models according to their average fit to the data; *i.e.*, by comparing the *marginal likelihoods* of competing models. Given two models, M_0 and M_1 , the Bayes factor comparison assessing the relative plausibility of each model as an explanation of the data, $BF(M_0, M_1)$, is:

$$BF(M_0, M_1) = \frac{\text{posterior odds}}{\text{prior odds}},$$

where odds = probability/(1 – probability) (Kass and Raftery, 1995). Accordingly, the posterior odds is the posterior probability of M_0 given the data, \mathbf{X} , divided by the posterior odds of M_1 given the data:

$$\text{posterior odds} = \frac{\mathbb{P}(M_0 | \mathbf{X})}{\mathbb{P}(M_1 | \mathbf{X})},$$

and the prior odds is the prior probability of M_0 divided by the prior probability of M_1 :

$$\text{prior odds} = \frac{\mathbb{P}(M_0)}{\mathbb{P}(M_1)}.$$

Thus, the Bayes factor measures the degree to which the data alter our belief regarding the support for M_0 relative to M_1 (Lavine and Schervish, 1999):

$$BF(M_0, M_1) = \frac{\mathbb{P}(M_0 | \mathbf{X})}{\mathbb{P}(M_1 | \mathbf{X})} \div \frac{\mathbb{P}(M_0)}{\mathbb{P}(M_1)}.$$

As in frequentist inference, the identification of the “best” model/hypothesis is somewhat subjective. That is, it’s up *you* to assess your degree of belief in M_0 relative to M_1 . Although there are no strict thresholds, the guidelines (proposed by Jeffreys, 1961) may be referred to when interpreting these measures (Table 2).

Table 1: Interpreting Bayes factors (Jeffreys, 1961).

$BF(M_0, M_1)$	Strength of evidence
$< 1 : 1$	Negative (supports M_1)
$1 : 1$ to $3 : 1$	Equivocal
$3 : 1$ to $10 : 1$	Substantial
$10 : 1$ to $30 : 1$	Strong
$30 : 1$ to $100 : 1$	Very strong
$> 100 : 1$	Decisive

AutoParts provides a flexible framework for evaluating various molecular evolutionary hypotheses regarding the nature of process heterogeneity using Bayes factors. For example, we may wish to test hypotheses regarding the number of distinct substitution processes in our sequence alignment. Alternatively, we may wish to assess whether two pre-specified data partitions share (or differ in) some aspect of the substitution process. We can test such hypotheses for various aspects of the substitution process, including the degree of ASRV (alpha-shape parameter), stationary frequencies, relative substitution rates (exchangeability rates), or the overall substitution rate (tree length). The flexibility of the hypothesis-testing/model-comparison framework stems from the ability to analytically calculate the relevant prior probability under the DPP model (using equations 1 and 2) and to efficiently estimate the corresponding posterior probability (using MCMC).

Imagine, for example, that we wish to evaluate the hypothesis that the conifer sequence exhibits two distinct overall substitution rates (tree lengths). For a given number of data partitions (here, $K = 6$) and for a specified value of the concentration parameter (here, $\alpha = 1.69577$), we can calculate the prior probability for two distinct tree-length parameters using eqn. 1. Recall that this prior is automatically calculated and reported in the post-analysis report log file, **conifer.slog**; $\mathbb{P} = 0.2619$, and the corresponding posterior probability estimate is $\mathbb{P} = 0.6007$ (Box 5). Accordingly, Bayes factor is:

$$BF(k = 2, k \neq 2) = \frac{\mathbb{P}(k = 2 \mid X, \alpha, K)}{1 - \mathbb{P}(k = 2 \mid X, \alpha, K)} \div \frac{\mathbb{P}(k = 2 \mid \alpha, K)}{1 - \mathbb{P}(k = 2 \mid \alpha, K)},$$

which equals ≈ 4.2 , suggesting that there is ‘substantial’ support for two distinct overall rates of substitution within the conifer alignment (Table 2).

Alternatively, we might be interested in assessing support for the hypothesis that data partitions for the third-position sites of the *atpB* and *rbcL* genes share the same set of stationary frequencies, π . The posterior probability that these two data partitions share the same process partition is reported in the post-analysis report; $\mathbb{P} = 0.1017$ (Box 5). We calculate the Bayes factor for this hypothesis as:

$$BF(\pi_3 = \pi_4, \pi_3 \neq \pi_4) = \frac{\mathbb{P}(\pi_3 = \pi_4 \mid X)}{1 - \mathbb{P}(\pi_3 = \pi_4 \mid X)} \times \alpha,$$

which equals ≈ 0.2 , indicating that there is no support for shared stationary frequencies within these two gene regions (Table 2), which is consistent with their grouping in separate process partitions for the stationary frequencies in the mean partition scheme (Figure 2).

Computational efficiency of AutoParts

When compared to other phylogenetic MCMC implementations, such as **MrBayes** or **BEAST**, **AutoParts** may appear painfully slow. For example, an analysis of the conifer dataset using **MrBayes** v. 3.2.2 (under a mixed model corresponding to the mean partition scheme, with a single chain run for 100,000 cycles using the beagle library; Suchard and Rambaut, 2009) required 81 seconds. By comparison, an analysis using **AutoParts** (under a fixed concentration parameter, run for 100,000 cycles) required ≈ 8 minutes. Accordingly, the analysis with **AutoParts** required almost 6 times longer to complete. However, this comparison ignores important differences in the algorithms and state space of these programs.

MCMC apples and oranges.—Most Bayesian phylogenetic programs, such as **MrBayes**, use a conventional Metropolis-Hastings version of Markov Chain Monte Carlo (Metropolis et al., 1953; Hastings, 1970) to approximate the joint posterior probability density of parameters. The Metropolis-Hastings algorithm is typically implemented such that a change is proposed to a single parameter during a cycle of the MCMC. By contrast, the MCMC algorithm implemented in **AutoParts** is a “Gibbs within Metropolis” MCMC algorithm, where many parameter updates are typically made during a single cycle of the simulation. Specifically, we use Algorithm 8 from Neal (2000) to propose updates to partition schemes. We first select a parameter to update according to the corresponding proposal probabilities; the default values in **AutoParts** are:

Table 2: Default proposal probabilities in AutoParts.

Parameter	Proposal probability
Tree topology, τ	0.44
Exchangeability rates, \mathbf{r}	0.14
Stationary frequencies, $\boldsymbol{\pi}$	0.14
ASRV parameter, α	0.14
Tree length, T	0.14

We have found that these proposal probabilities work well for many empirical analyses, and have hard coded them in **model.cpp**. However, you are free to modify these default values, if necessary, and then simply recompile the source code.

Imagine, for example, that we randomly selected the tree length parameter (with $\mathbb{P} = 0.14$), and that there are currently $k = 3$ process partitions occupied by the $K = 6$ data partitions with the allocation vector $\mathbf{z} = \{1, 1, 2, 2, 3, 3\}$. We set out a pre-specified number of *auxiliary tables* (we use $\kappa = 3$ auxiliary tables). The tree-length value for each of the auxiliary process partitions is then specified by making three independent draws from the corresponding prior (in this case, the gamma prior on tree length). We then propose an update to the allocation vector by evaluating the relative probabilities of all possible reassignments to the K data partitions to the k existing process partitions and the κ new auxiliary process partitions (Neal, 2000). Specifically, we select a data partition and remove it from its current process partition. If the data partition was the only member of the process partition, we then remove the process partition from computer memory. Otherwise, we decrease the count of the number of data partitions sharing that process partition by one (*e.g.*, decrease η_i).

We then calculate the likelihood of the data partition becoming a member of each of the m remaining process partitions in computer memory, \mathcal{L}_m . Also, calculate the likelihood of it becoming a member of one of the κ process partitions, \mathcal{L}_κ . The probability of the data partition becoming a member of the m^{th} process partition with η_m members is $\eta_m \times \mathcal{L}_m$. The probability of the data partition becoming a member of one of the κ auxiliary process partitions is $(\alpha/\kappa) \times \mathcal{L}_\kappa$. After the process partition joins a previously existing or a new (auxiliary) process partition, any unoccupied process partitions are deleted from memory. One MCMC cycle involves a scan of all K data partitions, with the above update mechanism applied to each. An additional update is performed to propose changes to the values of the tree lengths of every existing process partition.

From the above, it should be clear that a single MCMC cycle in **AutoParts** is typically equivalent to many cycles in other phylogenetic MCMC algorithms, which complicates direct comparisons of run times for a given number of ‘cycles’ between programs.

State space explored by AutoParts.—Analyses using programs such as **MrBayes** typically assume a single partition scheme, where MCMC simulation is used to estimate parameters of the assumed partition scheme, or stepping-stone simulation is used to estimate the marginal likelihood of the assumed partition scheme in order to compare it to competing mixed models using Bayes factors. By contrast, **AutoParts** estimates parameters of the phylogenetic model while averaging over all possible partition schemes in a single MCMC simulation.

The total number of process partitions for a single Dirichlet process with K data partitions is described by the Bell numbers (Bell, 1934). The Bell number for K elements is the sum of the Stirling numbers of the second kind:

$$\mathcal{B}(K) = \sum_{k=1}^n \mathcal{S}_2(K, k).$$

The Stirling number of the second kind, $\mathcal{S}_2(K, k)$, for K elements and k subsets (corresponding here to the number of data partitions and process partitions, respectively) is given by the following equation:

$$\mathcal{S}_2(K, k) = \frac{1}{k!} \sum_{i=0}^{k-1} (-1)^i \binom{k}{i} (k-i)^K.$$

Accordingly, the state space of possible process partitions quickly becomes large, even for alignments with relatively few data partitions. For example, each substitution model parameter for the conifer alignment, with $K = 6$ data partitions, has $\mathcal{B}(6) = 203$ possible process partitions. Because **AutoParts** implements an independent DPP model for each of the four substitution model parameters—base frequencies, exchangeability rates, tree length and α -shape parameter describing the degree of ASRV—there are $\mathcal{B}(K)^4 = 203^4 = 1,689,181,681$ possible partition schemes for the relatively simple conifer dataset.

A more apt comparison, therefore, would center on the time required to estimate the marginal likelihood of alternative partition schemes using a program such as **MrBayes**. If we could estimate marginal likelihoods at the rate of one partition scheme per second, it would require ≈ 53.8 years to evaluate all possible partition schemes for the conifer dataset. Note that this estimate is quite conservative; a stepping stone simulation using **MrBayes** (with a single chain drawing 100,000 samples from each of 49 stones) required 505 seconds to estimate the marginal likelihood of the mean partition scheme (coincidentally, this is approximately the time required for the **AutoParts** analysis). Viewed from this perspective, the computational efficiency of **AutoParts** appears more tolerable.

Even for relatively simple empirical problems, it seems that robust Bayesian model-selection methods (e.g., Xie et al., 2011; Fan et al., 2011) do not provide a practical solution for selecting among partition schemes. However, more efficient methods have been proposed to identify an optimal partition scheme. For example, **ParttiitonFinder** (Lanfear et al., 2012) uses likelihood-based model-selection criteria—the Akaike Information Criterion (AIC ; Akaike, 1974) and the Bayesian Information Criterion (BIC ; Kass and Raftery, 1995)—to quickly search for an optimal partition scheme. These metrics are based on the maximum-likelihood estimate, \mathcal{L} , for a candidate partition scheme. The AIC score is computed as $AIC = 2p_i - 2\ln \mathcal{L}$, where p_i is the number of free parameters in partition scheme i . Similarly, the BIC score is computed as $BIC = \ln n \times p_i - 2\ln \mathcal{L}$, where n is the number of independent observations and p_i is the number of free parameters in partition scheme i .

For example, a **PartitionFinder** analysis of the conifer alignment (using the ‘greedy’ algorithm to search the space of substitution models implemented in **MrBayes**) identified the following partition scheme as optimal under the AIC metric:

Box 6: Preferred partition scheme identified by **PartitionFinder** based on *AIC* metric.

Subset	Best Model	Subset Partitions	Subset Sites
1	GTR+I	atpB_1st	1-1394\3
2	GTR+G	atpB_2nd	2-1394\3
3	GTR+I	atpB_3rd	3-1394\3
4	HKY	rbcL_1st	1395-2659\3
5	GTR+G	rbcL_2nd	1396-2659\3
6	GTR	rbcL_3rd	1397-2659\3

For comparison, **PartitionFinder** using the *BIC* metric identified the following partition scheme as optimal:

Box 7: Preferred partition scheme identified by **PartitionFinder** based on *BIC* metric.

Subset	Best Model	Subset Partitions	Subset Sites
1	K80+I	atpB_1st	1-1394\3
2	GTR+G	atpB_2nd, rbcL_2nd	2-1394\3, 1396-2659\3
3	HKY+I	atpB_3rd	3-1394\3
4	K80	rbcL_1st, rbcL_3rd	1395-2659\3, 1397-2659\3

The alternative optimal partition schemes are markedly different: the *AIC*-based scheme has $k = 6$ distinct process partitions with 68 free parameters, whereas the *BIC*-based partition scheme has $k = 4$ distinct process partitions with 35 free parameters. It is unclear which partition scheme should be assumed for phylogeny estimation: there is no statistical theory to guide us in choosing between the *AIC* and the *BIC* (e.g., Kass and Raftery, 1995).

AutoParts and **PartitionFinder** differ fundamentally in the space of process partitions that they evaluate. First, **AutoParts** assumes GTR+ Γ as the base substitution model, whereas **PartitionFinder** evaluates a suite of candidate substitution models. Our modeling choice is motivated by theoretical results and simulation studies (e.g., Huelsenbeck and Rannala, 2004) demonstrating that—by virtue of demarginalizing estimates using MCMC—Bayesian inference is far more robust to inference under a mildly over-specified model than it is to bias caused by an underspecified model. In other words, we believe that it is preferable to focus our modeling efforts on capturing process heterogeneity under the most flexible possible model rather than worrying about inconsequential sub-models of the GTR, provided that we can achieve acceptable MCMC performance.

Second, **PartitionFinder** adopts an ‘all-or-none’ perspective on process heterogeneity. Consider, for example, the optimal partition scheme identified by **PartitionFinder** using the *BIC* metric (Box 7). Process partition 2 (comprising second-position sites for *atpB* and *rbcL*) are inferred to share *all* of the GTR+ Γ substitution model parameters, but share *none* of those substitution model parameters with other data partitions. By contrast, the mean partition scheme inferred using **AutoParts** indicates that—although these data partitions share common exchangeability rates and stationary frequencies—they nevertheless differ with respect to the tree length and degree of ASRV (Figure 2). Moreover, the exchangeability rates of this process partition are also shared by the first- and third-position sites of the *atpB* gene. The more granular perspective adopted by **AutoParts** allows partial sharing of individual substitution

model parameters between data partitions, which greatly enhances its ability to accommodate process heterogeneity.


Finally, the optimal partition scheme(s) identified by **PartitionFinder** become an inexorable assumption of the analysis. Our experiments with empirical datasets indicate that there is typically considerable uncertainty regarding the choice partition scheme. For example, our analysis of the conifer dataset indicates considerable uncertainty in the choice of partition scheme: 95% credible set includes 19 distinct partition schemes (the sum of the process partitions for the individual parameters). Failure to accommodate process heterogeneity is known to adversely impact phylogenetic inference, causing biased estimates of the tree topology and nodal support (Brandley, Schmitz and Reeder, 2005; Brown and Lemmon, 2007), estimates of branch lengths and divergence times (Marshall, Simon and Buckley, 2006; Poux et al., 2008; Vendetti, Meade and Pagel, 2008), and estimates of other model parameters (Nylander et al., 2004; Pagel and Meade, 2004). Accordingly, failure to accommodate *uncertainty in the choice of model* describing the process heterogeneity is likely to cause similarly biased estimates. For this reason, it is unwise to condition inference on *any* single partition scheme; even the ‘best’. Instead, we can make more robust phylogenetic inferences if we accommodate model uncertainty by averaging over all possible partition schemes.

Useful Links

- AutoParts: <http://brianrmoore.github.io/AutoParts/>
- bonsai: <https://bitbucket.org/mrmay/bonsai/overview>
- Homebrew: <http://brew.sh>
- Tree Thinkers: <http://treethinkers.org>

Questions about this tutorial can be directed to:

- Brian R. Moore (email: brianmoore@ucdavis.edu)
- John P. Huelsenbeck (email: johnh@ucberkeley.edu)

 This tutorial was written by Brian Moore and John Huelsenbeck. We are indebted to Mike May for writing the **AutoPlots.R** graphing tools, to Sebastian Höhna for optimizing the code, and Tracy Heath for helpful comments and feedback. This research was supported by grants from the NSF (DEB-0445453) and NIH (GM-069801) awarded to J.P.H., and by NSF grants (DEB-0842181, DEB-0919529, and DBI-1356737) awarded to BRM. Computational resources for this work were provided by an NSF XSEDE grant (DEB-120031) to BRM. This guide is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

Version dated: September 9, 2014

References

- Akaike H. 1974. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*. 19:716–723.
- Bell ET. 1934. Exponential numbers. *American Mathematics Monthly*. 41:411–419.
- Brandley MC, Schmitz A, Reeder TW. 2005. Partitioned Bayesian analyses, partition choice, and the phylogenetic relationships of scincid lizards. *Systematic Biology*. 54:373–390.
- Brown JM, Lemmon AR. 2007. The importance of data partitioning and the utility of Bayes factors in Bayesian phylogenetics. *Systematic Biology*. 56:643–655.
- Fan Y, Wu R, Chen MH, Kuo L, Lewis PO. 2011. Choosing among partition models in Bayesian phylogenetics. *Molecular Biology and Evolution*. 28:523–532.
- Gusfield D. 2002. Partition-distance: a problem and class of perfect graphs arising in clustering. *Information Processing Letters*. 82:159–164.
- Hastings WK. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*. 57:97–109.
- Huelsenbeck JP, Rannala B. 2004. Frequentist properties of Bayesian posterior probabilities of phylogenetic trees under simple and complex substitution models. *Systematic Biology*. 53:904–913.
- Jeffreys H. 1961. *Theory of Probability*. Oxford: Oxford University Press.
- Kass RE, Raftery AE. 1995. Bayes factors. *Journal of the American Statistical Association*. 90:773–795.
- Lanfear R, Calcott B, Ho SYW, Guindon S. 2012. Partitionfinder: Combined selection of partitioning schemes and substitution models for phylogenetic analyses. *Molecular Biology and Evolution*. 29:1695–1701.
- Lartillot N, Philippe H. 2006. Computing Bayes factors using thermodynamic integration. *Systematic Biology*. 55:195–207.
- Lavine M, Schervish MJ. 1999. Bayes factors: What they are and what they are not. *American Statistician*. 53:119–122.
- Marshall DC, Simon C, Buckley TR. 2006. Accurate branch length estimation in partitioned Bayesian analyses requires accommodation of among-partition rate variation and attention to branch length priors. *Systematic Biology*. 55:993–1003.
- Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. 1953. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*. 21:1087–1092.
- Neal RM. 2000. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*. 9:249–265.
- Newton MA, Raftery AE. 1994. Approximate Bayesian inference by the weighted likelihood bootstrap. *Journal of the Royal Statistical Society, B*. 56:3–48.
- Nylander JAA, Ronquist F, Huelsenbeck JP, Aldrey JLN. 2004. Bayesian phylogenetic analysis of combined data. *Systematic Biology*. 53:47–67.

- Pagel M, Meade A. 2004. A phylogenetic mixture model for detecting pattern-heterogeneity in gene sequence or character-state data. *Systematic Biology*. 53:571–581.
- Poux C, Madsen O, Glos J, de Jong WW, Vences M. 2008. Molecular phylogeny and divergence times of Malagasy tenrecs: Influence of data partitioning and taxon sampling on dating analyses. *BMC Evolutionary Biology*. 8:102.
- Raftery A, Newton M, Satagopan J, Krivitsky P. 2007. Estimating the integrated likelihood via posterior simulation using the harmonic mean identity. In: Bernardo JM, Bayarri MJ, Berger JO, editors, *Bayesian statistics*. Oxford University Press, pp. 1– 45.
- Rambaut A, Drummond AJ. 2007. Tracer v1.5. <http://beast.bio.ed.ac.uk/Tracer>.
- Stanton D, White D. 1986. *Constructive Combinatorics*. New York: Springer-Verlag.
- Suchard MA, Rambaut A. 2009. Many-core algorithms for statistical phylogenetics. *Bioinformatics*. 25:1370–1376.
- Suchard MA, Weiss RE, Sinsheimer JS. 2001. Bayesian selection of continuous-time Markov chain evolutionary models. *Molecular Biology and Evolution*. 18:1001–1013.
- Tierney L. 1994. Markov chains for exploring posterior distributions (with discussion). *Annals of Statistics*. 22:1701–1762.
- Vendetti C, Meade A, Pagel M. 2008. Phylogenetic mixture models can reduce node-density artifacts. *Systematic Biology*. 58:286–293.
- Xie W, Lewis PO, Fan Y, Kuo L, Chen MH. 2011. Improving marginal likelihood estimation for Bayesian phylogenetic model selection. *Systematic Biology*. 60:150–160.