# PartitionFinder: Combined Selection of Partitioning Schemes and Substitution Models for Phylogenetic Analyses

Robert Lanfear,*[,1] Brett Calcott,[1,2] Simon Y. W. Ho,[3] and Stephane Guindon[4]

[1]Centre for Macroevolution and Macroecology, Ecology Evolution and Genetics, Research School of Biology, Australian National University, Canberra, Australian Capital Territory, Australia

[2]Philosophy Program, Research School of Social Sciences, Australian National University, Canberra, Australian Capital Territory, Australia

[3]School of Biological Sciences, University of Sydney, Sydney, New South Wales, Australia

[4]Department of Statistics, University of Auckland, Auckland, New Zealand

***Corresponding author:** E-mail: rob.lanfear@anu.edu.au.

**Associate editor:** Sudhir Kumar

## Abstract

In phylogenetic analyses of molecular sequence data, partitioning involves estimating independent models of molecular evolution for different sets of sites in a sequence alignment. Choosing an appropriate partitioning scheme is an important step in most analyses because it can affect the accuracy of phylogenetic reconstruction. Despite this, partitioning schemes are often chosen without explicit statistical justification. Here, we describe two new objective methods for the combined selection of best-fit partitioning schemes and nucleotide substitution models. These methods allow millions of partitioning schemes to be compared in realistic time frames and so permit the objective selection of partitioning schemes even for large multilocus DNA data sets. We demonstrate that these methods significantly outperform previous approaches, including both the ad hoc selection of partitioning schemes (e.g., partitioning by gene or codon position) and a recently proposed hierarchical clustering method. We have implemented these methods in an open-source program, PartitionFinder. This program allows users to select partitioning schemes and substitution models using a range of information-theoretic metrics (e.g., the Bayesian information criterion, akaike information criterion [AIC], and corrected AIC). We hope that PartitionFinder will encourage the objective selection of partitioning schemes and thus lead to improvements in phylogenetic analyses. PartitionFinder is written in Python and runs under Mac OSX 10.4 and above. The program, source code, and a detailed manual are freely available from www.robertlanfear.com/partitionfinder.

**Key words:** partitioning, AIC, BIC, AICc, model selection, molecular evolution.

## Introduction

Molecular phylogenetics provides a wealth of important information for evolutionary biologists. However, the accuracy of molecular phylogenetic inference depends on having an appropriate model of molecular evolution (Sullivan and Joyce 2005; Simon et al. 2006). Because of this, there is a great deal of interest in developing methods to select evolutionary models and assess their adequacy (Ripplinger and Sullivan 2010; Jayaswal et al. 2011; Nguyen et al. 2011). The goal of model selection is to identify a model that is sufficiently complex to capture the evolutionary processes that have occurred but to avoid models with more parameters than can be reliably estimated from the available data (overparameterization). One of the most important aspects of models of molecular evolution is how they account for variation in evolutionary processes among the sites of an alignments, because the failure to correctly account for this variation can seriously mislead phylogenetic analyses (Buckley et al. 2001; Telford and Copley 2011).

There are two ways to incorporate the variation in evolutionary processes among different sites using currently available phylogenetic methods: mixture models and partitioning. With mixture models, the likelihood of each site is calculated under more than one substitution model (e.g., Le et al. 2008). The parameters of these substitution models, as well as the probability with which each model applies to each site, can be determined directly from the data (Pagel and Meade 2004). With partitioning, the user first groups together sites that are assumed to have evolved under similar processes and then estimates independent (i.e., unlinked) substitution models for each group of sites (e.g., Nylander et al. 2004; Brandley et al. 2005; McGuire et al. 2007). In contrast to mixture models, partitioning requires the a priori definition of appropriate groups of sites. Although mixture models are implemented in an increasing variety of phylogenetic software (e.g., Pagel and Meade 2004; Stamatakis 2006; Le et al. 2008), partitioning remains by far the most common approach to incorporating heterogeneity in evolutionary processes among sites (Blair and Murphy 2011).

Choosing an appropriate partitioning scheme is a central problem for most phylogenetic analyses (Brandley et al. 2005; Shapiro et al. 2006; McGuire et al. 2007; Li et al. 2008; Blair and Murphy 2011). Typically, phylogeneticists use their biological intuition to group together similar sites in an alignment into putatively homogeneous data blocks.

This often involves defining data blocks on the basis of genes and codon positions (e.g., Shapiro et al. 2006; Ho and Lanfear 2010). For example, in an analysis of four protein-coding genes, one could define 12 data blocks—one for each codon position in each gene. This approach is biologically justified because differences between codon positions and genes are expected to account for much of the heterogeneity in evolutionary processes among sites (Shapiro et al. 2006). However, many studies have shown that this approach can lead to overparameterization, and that phylogenetic reconstruction can be improved by merging certain data blocks together, thus defining a partitioning scheme that requires the estimation of fewer independent substitution models (Brandley et al. 2005; Brown and Lemmon 2007; McGuire et al. 2007; Li et al. 2008). For example, the second codon positions in two similar nuclear genes may experience similar rates and patterns of substitution and so might be better analyzed together rather than independently. Of course, it is not always straightforward to identify which data blocks should be merged and which should be analyzed independently. One solution to this problem is to compare all possible partitioning schemes for a given data set. However, this approach is usually computationally intractable because the number of possible partitioning schemes is astronomical even for relatively small numbers of data blocks (Li et al. 2008). As a result, most researchers either choose a single partitioning scheme a priori or select the best-fit scheme from a handful of candidate schemes (Brandley et al. 2005; McGuire et al. 2007). Thus, despite significant advances in phylogenetic methods in recent years, the accuracy of the inferences we can make from partitioned phylogenetic analyses remains limited by our ability to select appropriate partitioning schemes.

In this study, we describe two new methods that solve many of the problems associated with selecting partitioning schemes. These methods increase the efficiency of comparing partitioning schemes by many orders of magnitude, allowing many millions of schemes to be compared in realistic time frames. We describe these new methods below and assess their performance on a range of published data sets. We show that our methods select significantly better partitioning schemes than previous approaches—including the ad hoc selection of partitioning schemes and previously suggested objective approaches. We have implemented these methods in an open-source program, PartitionFinder. This program has flexible options and allows users to efficiently and objectively find best-fit partitioning schemes and nucleotide substitution models, even for large data sets. PartitionFinder, its source code, and a detailed manual are available from www.robertlanfear.com/partitionfinder.

## Materials and Methods

We use the following definitions throughout this article. We define a "data block" as a user-defined set of sites in an alignment; a "subset" as a set of one or more data blocks; and a partitioning scheme as a set of subsets that includes all sites in the alignment once and only once. For clarity, we avoid the use of the term "partition," as this has different and potentially very confusing meanings in the mathematical and molecular phylogenetics literature (in the mathematical literature, a partition is equivalent to our use of "partitioning scheme" here, whereas in the molecular phylogenetics literature, it is equivalent to our use of "subset" here). In the majority of cases, users will specify data blocks based on genes and codon positions—for example, by defining 12 data blocks for an alignment of four protein-coding genes. The sites in a data block need not be contiguous in the alignment, but a single site can be a member of only one data block. A subset can comprise a single data block (e.g., first codon sites from a protein-coding gene) or multiple data blocks (e.g., first and second codon sites from a protein-coding gene). For example, consider an alignment of four protein-coding genes for which the user has defined 12 data blocks, one for each codon position in each gene. One possible partitioning scheme for this data set involves treating each codon position in each gene independently. This partitioning scheme has 12 subsets, and so 12 unlinked substitution models would be estimated from the data during the phylogenetic analysis. Another possible partitioning scheme involves treating each codon position independently but merging the codon positions across genes. This partitioning scheme has three subsets (one for each codon position), and so three unlinked substitution models would be estimated from the data during the phylogenetic analysis. The challenge is to find the best-fit partitioning scheme for a given nucleotide alignment, given the predefined set of data blocks.

The number of possible partitioning schemes for a set of $n$ data blocks is equivalent to the number of ways of putting $n$ different-colored balls into one or more indistinguishable boxes. This relationship is known as a Bell number (Bell 1934) and can be described by the following relationship, where $B_n$ is the number of possible partitioning schemes given $n$ user-defined data blocks (Li et al. 2008), and the curly brackets define a Stirling number of the second kind:

$$B_n = \sum_{k=0}^{n} \left\{ {n \atop k} \right\}.$$

The number of possible partitioning schemes can be astronomical even for relatively modest data sets. For example, in an analysis of four protein-coding genes (4 genes $\times$ 3 codons $=$ 12 data blocks), there are $B_{12} = 4.2 \times 10^6$ possible partitioning schemes, and for an analysis of 20 protein-coding genes (20 genes $\times$ 3 codons $=$ 60 data blocks), there are $B_{60} = 9.8 \times 10^{59}$ possible partitioning schemes.

The set of partitioning schemes will be made up of a smaller number of possible subsets because most subsets will be included in a many different partitioning schemes. Specifically, the number of possible subsets, $S_n$, that can be created from a set of $n$ user-defined data blocks is the

number of possible nonempty subsets that can be generated from a set of size $n$:

$$S_n = 2^n - 1.$$

For example, in an analysis of four protein-coding genes (12 data blocks), there are $S_{12} = 4{,}095$ possible subsets, and in an analysis of 20 protein-coding genes (60 data blocks), there are $S_{60} = 1.2 \times 10^{18}$ possible subsets.

## The PartitionFinder Algorithm

Previous approaches to comparing partitioning schemes have been both labor-intensive and computationally intensive because they have required a full likelihood or Bayesian analysis for each partitioning scheme under consideration (see e.g., McGuire et al. 2007; Li et al. 2008). This has fundamentally limited the number of partitioning schemes that have been compared in most studies, as comparing large numbers (e.g., hundreds) of partitioning schemes in this way is simply not feasible for most data sets. This approach is also highly inefficient because it involves repeatedly recalculating the likelihood of every site in the alignment, despite the fact that the substitution models applied to those sites will be the same for many partitioning schemes. The PartitionFinder algorithm improves the efficiency of finding best-fit partitioning schemes by calculating the log likelihood of each subset of sites only once. The log likelihood of each partitioning scheme is then calculated by summing the log likelihoods of the subsets that make up that scheme.

An outline of the PartitionFinder algorithm is as follows:

1. Estimate a phylogenetic tree of sequences;
2. Select the best-fit substitution model for each possible subset;
3. Calculate the log likelihood of each partitioning scheme by summing the log likelihoods of the subsets that make up that scheme;
4. Select a partitioning scheme using information-theoretic metrics.

All likelihood calculations are performed using a modified version of PhyML 3.0 (Guindon et al. 2010), available from the authors and as part of the PartitionFinder program. Tree estimation (step 1) is performed using the BioNJ algorithm implemented in PhyML 3.0 (Guindon et al. 2010), using the combined data from all of the user-defined data blocks. PartitionFinder also allows the user to specify a tree topology for step 1. The tree topology from step 1 is then fixed for the rest of the analysis. This differs from previous approaches, which coestimate the tree topology and the likelihood of each partitioning scheme. This is a computationally intensive method that has limited the number of partitioning schemes that can be compared (see above). Using a fixed tree topology allows likelihoods from different subsets to be combined, which increases the efficiency by many orders of magnitude and allows many millions of partitioning schemes to be compared in a single run. Fixing the tree topology is unlikely to adversely affect the results of comparing partitioning schemes, as previous studies have shown that doing so does not affect the results of model selection procedures as long as a nonrandom tree topology is used (Posada and Crandall 2001).

Model selection (step 2) is performed on a user-specified set of up to 56 substitution models from the general time reversible (GTR) family, and our approach is similar to other model selection algorithms (e.g., Keane et al. 2006; Posada 2008). During model selection, we first calculate the likelihood of each candidate substitution model, conditioned on the tree topology from step 1. We then select the best-fit model according to one of three user-specified information-theoretic metrics: the akaike information criterion (AIC), the corrected AIC (AICc), or the Bayesian information criterion (BIC) (Sullivan and Joyce 2005). PartitionFinder implements almost all of the models of nucleotide evolution included in the most commonly used phylogenetic tree estimation programs such as PhyML (Guindon et al. 2010), RaxML (Stamatakis 2006), MrBayes (Ronquist and Huelsenbeck 2003), and BEAST (Drummond and Rambaut 2007). This means that the output from PartitionFinder can be used to directly set up a phylogenetic analysis in any of these programs. However, all of these models and programs assume that the data evolved under a time-reversible, stationary, and homogeneous process, and they should not be used if the data violate any of these assumptions.

PartitionFinder includes an option for either linked or unlinked branch lengths between subsets. When branch lengths are linked, step 1 includes the reestimation of branch lengths on the BioNJ topology using a GTR substitution model, with a proportion of invariant sites and gamma distributed rates across sites estimated from the data. The likelihood of each model for each subset (step 2) is then calculated conditioned on this topology and these branch lengths, with each model afforded an independent rate multiplier that can increase or decrease all branch lengths by the same factor. Thus, linked branch lengths allow for subset-specific substitution rates, but all subsets share a single set of relative branch lengths. By contrast, when branch lengths are unlinked, model selection (step 2) is conditioned on the topology from step 1, but all branch lengths are estimated independently for each model in each subset.

The log likelihood of each partitioning scheme (step 3) is calculated by summing the log likelihoods of the best-fit model for each subset in the partitioning scheme. Finally, the best-fit partitioning scheme is selected (step 4) using one of three information-theoretic measures: the AIC, AICc, or BIC.

## A Greedy Heuristic Algorithm to Search for Partitioning Schemes

Even using the algorithm described above, exhaustive searches on desktop computers are practically limited to data sets for which 12 or fewer data blocks are defined (corresponding to data sets with 4.2 million or fewer possible partitioning schemes). Therefore, heuristic searches

among partitioning schemes are necessary for larger data sets, even though they cannot be guaranteed to find the optimum partitioning scheme (Li et al. 2008).

The heuristic search algorithm we describe below incorporates the increases in efficiency described above but hugely reduces the number of partitioning schemes that need to be considered for a given data set. Our method builds on a recently proposed method (Li et al. 2008) that involves estimating GTR+G model parameters for each data block and then progressively merging the data blocks with the most similar parameter estimates using hierarchical cluster analysis. For a set of $n$ data blocks, the hierarchical clustering method objectively defines $n$ partitioning schemes that range from having $n$ subsets (all data blocks treated independently) to having a single subset (all data blocks merged together). The optimal scheme is then selected from this set of $n$ schemes using an information-theoretic metric (e.g., the AIC, AICc, or BIC).

Because the hierarchical clustering approach combines data blocks based on model parameter estimates, it relies on those parameter estimates being accurate. For many data blocks, there will be limited information available for estimating many of the GTR+G model parameters. This will result in these estimates being associated with high variance because the value of the parameters will have very little effect on the overall likelihood score. Since the subsequent hierarchical clustering method treats all parameters as equally important, uncertain parameter estimates might limit the ability of the hierarchical clustering approach to find optimal partitioning schemes. The algorithm we propose below overcomes this limitation by merging data blocks based directly on information-theoretic comparisons between partitioning schemes. These metrics are calculated directly from the likelihood so they implicitly incorporate the relative importance of different model parameters and so avoid problems associated with error-prone parameter estimates.

In an analysis with $n$ data blocks, our greedy heuristic algorithm begins by calculating the information-theoretic score (e.g., AIC, AICc, or BIC) of the partitioning scheme with $n$ subsets, that is, the scheme in which each data block is treated independently ($P_{start}$). It then calculates the score of all partitioning schemes with $n - 1$ subsets, that is, all schemes that merge two subsets of $P_{start}$, and selects the scheme with the best score ($P_{merged}$). If $P_{merged}$ has a better score than $P_{start}$, $P_{merged}$ replaces $P_{start}$, and the algorithm iterates. The algorithm continues until either $P_{merged}$ does not have a better score than $P_{start}$, or until all data blocks have been merged into one subset. This process results in a greedy hill-climbing algorithm that optimizes the information-theoretic score of interest while searching for partitioning schemes.

We can calculate the maximum number of partitioning schemes ($P_{n\_greedy}$) that would need to be examined by this algorithm as follows. In addition to the starting scheme, each round of the algorithm involves calculating the likelihood of $k$ choose two schemes, where $k$ is the number of subsets in the best scheme from the previous round. In the worst case, the algorithm has to continue until $k = 2$, at which point the partitioning scheme under consideration has all data blocks merged into one subset. Thus, in an analysis with $n$ data blocks, the maximum number of partitioning schemes $P_{n\_greedy}$ considered by this algorithm is:

$$P_{n\_greedy} = 1 + \sum_{k=2}^{n} \binom{k}{2} = 1 + n(n^2 - 1)/6.$$

The maximum number of subsets that need to be examined by this algorithm ($S_{n\_greedy}$) is smaller than the maximum number of partitioning schemes because many subsets are contained in more than one scheme. $S_{n\_greedy}$ can be calculated as follows. The starting scheme involves examining $n$ subsets. In the next round of the algorithm, we examine all $n$ choose two subsets that merge two data blocks of the starting scheme. In subsequent rounds, we need only examine the $k - 2$ novel subsets that can be created by merging the most recently created subset with the remaining subsets in the current partitioning scheme. Thus, the maximum number of subsets that need to be considered by this algorithm is:

$$S_{n\_greedy} = n^2 - n + 1.$$

The greedy algorithm can be many orders of magnitude more efficient than an exhaustive search. For instance, a data set with 60 data blocks requires the analysis of $B_{60} = 9.77 \times 10^{59}$ partitioning schemes and $S_{60} = 1.15 \times 10^{18}$ subsets for an exhaustive search, but at most $P_{60\_greedy} = 35{,}991$ partitioning schemes and $S_{60\_greedy} = 3{,}541$ subsets with the heuristic algorithm described here.

## Comparing Exhaustive and Heuristic Searches in PartitionFinder

We tested the ability of our heuristic algorithm to find optimal partitioning schemes for ten data sets obtained from Data Dryad (www.datadryad.org) and TreeBase (www.treebase.org; table 1). The data sets we used range from 13 to 164 taxa, from 1,896 to 9,005 bp, and from 6 to 12 data blocks (table 1). They include a range of introns, protein-coding genes, and RNA genes from the mitochondrial and nuclear genomes and are typical of the multilocus data sets routinely used for phylogenetic analyses.

For each nucleotide sequence alignment (table 1), we excluded sites that had been excluded by the authors of the original study and then defined data blocks based on genes and codon positions, treating transfer RNAs (tRNAs) as a single data block. For some data sets, we excluded certain genes used in the original studies in order to limit the size of each data set to a maximum of 12 data blocks, thus permitting an exhaustive search of partitioning schemes. To find the optimal partitioning scheme, we used the algorithm described above, implemented in PartitionFinder, to perform an exhaustive search of all possible partitioning schemes on each data set. We then used

**Table 1.** Properties of the Ten Data Sets Used to Compare Different Approaches to Selecting Partitioning Schemes.

| Taxon | Reference | Number of Spp. | Sites | Loci Used (* denotes non–protein-coding) | Data Blocks | Number of Possible Partitioning Schemes |
|---|---|---|---|---|---|---|
| Moths | Mitchell et al. (2000) | 77 | 1,949 | DDC, EF1α | 6 | 203 |
| Bark beetles | Cognato and Vogler (2001) | 44 | 1,896 | COI, EFIα, 16S* | 7 | 877 |
| Swallowtail butterflies | Caterino et al. (2001) | 37 | 3,228 | COI, COII, EF1α | 9 | 21,147 |
| Rodents | Huchon et al. (2002) | 42 | 3,633 | A2AB, IRBP, vWF | 9 | 21,147 |
| Hummingbirds | McGuire et al. (2007) | 164 | 3,821 | ND2, ND4, Bfib*, AKI*, tRNA* | 9 | 21,147 |
| Skinks | Miralles et al. (2011) | 33 | 3,936 | BDNF, C-mos, α-Enolase | 9 | 21,147 |
| Midges | Ekrem et al. (2010) | 74 | 2,701 | COI, COII, CAD, 16S* | 10 | 115,975 |
| Saxifragales (Eudicots) | Fishbein et al. (2001) | 40 | 9,005 | atpB, matK, rbcL, 18S*, 26S* | 11 | 678,570 |
| Clearwing butterflies | Elias et al. (2009) | 143 | 4,159 | COI, COII, EFIα, tetkin | 12 | 4,213,597 |
| Armadillos | Delsuc et al. (2003) | 13 | 6,070 | ADRA2B, BRCA1, vWF, ND1 | 12 | 4,213,597 |

PartitionFinder to perform a heuristic search on each data set using the heuristic algorithm described above and asked whether the heuristic search was able to find the optimal partitioning scheme for each data set. For all analyses, branch lengths were linked between subsets, all 56 available substitution models were considered for each subset, and substitution model selection and partitioning scheme selection were carried out using the BIC. All input files are available from the authors or from www.datadryad.org.

## Comparing Partitioning Schemes Selected by PartitionFinder to Commonly Used A Priori Partitioning Schemes

For each data set in table 1, we compared the optimal partitioning scheme with four commonly used a priori schemes: 1) no partitioning (i.e., all data treated as a single subset); 2) data partitioned by gene; 3) data partitioned by codon position; and 4) data partitioned by gene and codon position (see table 2).

We used the user-defined search option in PartitionFinder to calculate the BIC score of each a priori partitioning scheme for each data set. When partitioning by codon position, we treated codon positions of protein-coding genes from the mitochondrial and nuclear genomes as independent subsets and partitioned all other data blocks (e.g., tRNAs, ribosomal RNAs, and introns) by gene. All other settings in PartitionFinder were as described above.

## Larger Data Sets: Comparing Partitioning Schemes Selected by PartitionFinder with Those Selected by Hierarchical Clustering

The hierarchical clustering approach described by Li et al. (2008) is both computationally intensive and labor-intensive and has to date been applied to only a single data set. This data set comprises ten nuclear protein-coding genes (i.e., 30 data blocks) from 72 ray-finned fish, totaling 7,995 bp (Li et al. 2008). The hierarchical clustering method was applied to this data set to select optimal partitioning schemes in four different ways (Li et al. 2008): by estimating the GTR+G parameters using Maximum likelihood (ML) and selecting a partitioning scheme based on either the AIC ($HC_{AIC\_ML}$, table 3) or the BIC ($HC_{BIC\_ML}$, table 3); and by estimating the GTR+G parameters using Bayesian inference and selecting a partitioning scheme based on the either the BIC ($HC_{BIC\_Bayes}$, table 3) or the Bayes factors ($HC_{BF\_Bayes}$).

We used the heuristic algorithm implemented in PartitionFinder to select partitioning schemes for this data set using both the AIC ($PF_{AIC}$, table 3) and the BIC ($PF_{BIC}$, table 3), with other settings as described above. We then compared all six partitioning schemes (four selected using hierarchical clustering and two using PartitionFinder) by optimizing the tree topology under each partitioning scheme using RAxML v7.2.8 with ten independent topology search replicates for each partitioning scheme and a separate GTR+G model applied to each subset

**Table 2.** A Comparison of Partitioning Schemes Selected Using PartitionFinder and A Priori Approaches.

| Data Set | Data Blocks | Optimum Partitioning Scheme (BIC) | PartitionFinder Search (ΔBIC) | No Partitioning (ΔBIC) | Partitioned by Gene (ΔBIC) | Partitioned by Codon Position (ΔBIC) | Partitioned by Gene and Codon Position (ΔBIC) |
|---|---|---|---|---|---|---|---|
| Moths | 6 | 65,903.2 | 0 | −1,775.8 | −1,541.6 | −477.5 | 0.0 |
| Bark beetles | 7 | 37,911.9 | 0 | −1,853.5 | −1,251.3 | −468.2 | −15.9 |
| Swallowtail butterflies | 9 | 63,152.8 | 0 | −4,302.0 | −3,429.9 | −1.9 | −127.4 |
| Rodents | 9 | 112,900.0 | 0 | −2,165.0 | −2,102.9 | −91.1 | −125.4 |
| Hummingbirds | 9 | 185,747.1 | 0 | −6,610.8 | −2,809.4 | −29.2 | −59.5 |
| Skinks | 9 | 9,726.2 | 0 | −195.7 | −128.7 | −132.5 | −98.9 |
| Midges | 10 | 82,716.2 | 0 | −4,190.6 | −2,647.6 | −64.2 | −52.2 |
| Saxifragales (Eudicots) | 11 | 88,684.2 | 0 | −2,814.6 | −1,208.1 | −818.1 | −253.3 |
| Clearwing butterflies | 12 | 45,092.8 | 0 | −3,608.8 | −1,898.1 | −108.6 | −149.9 |
| Armadillos | 12 | 45,828.0 | 0 | −3,383.0 | −1,545.9 | −1,215.6 | −111.4 |
| Mean ΔBIC | | | 0 | −3,090.0 | −1,856.4 | −340.7 | −99.4 |

NOTE.—The optimum partitioning scheme for each data set was found using the exhaustive search feature in PartitionFinder. The difference in the BIC score between the optimum scheme and the five other partitioning schemes (ΔBIC) is shown. Details of data sets are provided in table 1. All BIC scores were calculated in PartitionFinder.

**Table 3.** A Comparison of Partitioning Schemes Selected Using PartitionFinder and the Hierarchical Clustering Approach of Li et al. (2008).

| Name | Selection Method | Subsets | Parameters | AIC | BIC |
|---|---|---|---|---|---|
| $HC_{BIC\_ML}$ | Hierarchical clustering | 10 | 191 | 253,472.6 | 254,807.0 |
| $HC_{BF\_Bayes}$ | Hierarchical clustering | 17 | 254 | 253,076.4 | 254,851.0 |
| $HC_{BIC\_Bayes}$ | Hierarchical clustering | 22 | 299 | 252,787.4 | 254,876.4 |
| $HC_{AIC\_ML}$ | Hierarchical clustering | 30 | 371 | 252,667.4 | 255,259.4 |
| $PF_{AIC}$ | PartitionFinder search | 27 | 344 | 252,549.7 | 254,953.1 |
| $PF_{BIC}$ | PartitionFinder search | 12 | 209 | 252,816.1 | 254,276.3 |

NOTE.—The best-scoring partitioning scheme for the AIC and the BIC is underlined. The method used to choose each partitioning scheme is described in the text.

(Stamatakis 2006). Finally, we calculated the AIC and BIC scores of each of the six partitioning schemes based on the likelihoods of the ML topology estimated in RAxML. All analysis files are available from the authors or from www.datadryad.org.

## Results and Discussion

### Heuristic Searches in PartitionFinder Find Optimal Partitioning Schemes

The heuristic algorithm implemented in PartitionFinder was able to find the optimal partitioning scheme in all ten of the data sets that we examined (table 2). For the moth data set, this result is trivial because the optimal scheme involves treating all data blocks independently, and the heuristic algorithm is guaranteed to discover this scheme because it is used as the starting scheme for the heuristic search. In all other cases, however, the optimal scheme had at least two fewer subsets than there were data blocks. These results suggest that the heuristic search in PartitionFinder is a reliable method of selecting best-fit partitioning schemes. This heuristic approach will be of particular use for large data sets, for which exhaustive searches are not feasible.

### A Priori Approaches to Choosing Partitioning Schemes Are Usually Suboptimal, but Some Methods Are Better than Others

Our results demonstrate that commonly used a priori partitioning schemes are rarely optimal and are often severely over- or underparameterized (table 2). From the ten data sets we examined, there was only a single case in which the optimal scheme was selected a priori: when partitioning by gene and codon position with the moth data set (table 2). For all other cases, a priori partitioning schemes performed much worse than either exhaustive or heuristic searches in PartitionFinder. This highlights the utility of methods such as those presented here, which allow very large numbers of partitioning schemes to be compared and for the best scheme to be selected objectively.

Of the four a priori approaches to partitioning that we compared, not partitioning at all resulted in the worst BIC scores (on average 3,090 BIC units worse than the optimal scheme, table 2), followed by partitioning by gene (on average 1,856 BIC units worse than the optimal scheme), partitioning by codon position (on average 341 BIC units worse than the optimal scheme), and finally partitioning by gene and codon position (on average 99 BIC units worse than the optimal scheme). These results highlight that a failure to partition the data at all or partitioning it by gene only (which remains surprisingly common in molecular phylogenetic analyses) can result in highly suboptimal partitioning schemes and may severely limit the accuracy of phylogenetic analyses in some cases. Our results suggest that in the absence of objective comparisons of large numbers of partitioning schemes, the most reliable ad hoc approach is to partition on the basis of gene and codon position, although even this approach can be highly suboptimal for some data sets (table 2).

### Heuristic Searches in PartitionFinder Outperform Hierarchical Clustering

The heuristic algorithm implemented in PartitionFinder selected better partitioning schemes than a recently proposed hierarchical clustering approach on a ten gene data set from ray-finned fishes (table 3; Li et al. 2008). The best scheme selected by PartitionFinder was better than the best scheme selected using hierarchical clustering for both the AIC (118 units difference; table 3) and the BIC (531 units difference; table 3). These improvements are large and suggest that our heuristic algorithm is able to overcome some of the limitations of the hierarchical clustering approach (see above).

## Conclusions

The methods we have presented increase the efficiency of comparing and searching for partitioning schemes by many orders of magnitude. We have demonstrated that they routinely outperform other ad hoc and objective methods for choosing partitioning schemes. The implementation of these methods in freely available open-source software paves the way for their routine use in molecular phylogenetics. Our analyses demonstrate that PartitionFinder can be used to compare millions of partitioning schemes in a single run and to select good and often optimal partitioning schemes for a large range of DNA data sets. We hope that PartitionFinder will simplify the selection of partitioning schemes and lead to concomitant improvements in phylogenetic analyses.

## Acknowledgments

We thank Matt Brandley for helpful comments on the manuscript, and Ainslie Seago, Renee Catullo, Jess Thomas,

and Matt Brandley for helpful comments on PartitionFinder. This work was supported by the Australian Research Council (R.L., B.C., S.Y.W.H.), and by the Marsden Fund (S.G.), project 08-UOA-068.

## References

Bell ET. 1934. Exponential numbers. *Am Math Mon.* 41:411–419.

Blair C, Murphy RW. 2011. Recent trends in molecular phylogenetic analysis: where to next? *J Hered.* 102:130–138.

Brandley MC, Schmitz A, Reeder TW. 2005. Partitioned Bayesian analyses, partition choice, and the phylogenetic relationships of scincid lizards. *Syst Biol.* 54:373–390.

Brown JM, Lemmon AR. 2007. The importance of data partitioning and the utility of Bayes factors in Bayesian phylogenetics. *Syst Biol.* 56:643–655.

Buckley TR, Simon C, Chambers GK. 2001. Exploring among-site rate variation models in a maximum likelihood framework using empirical data: effects of model assumptions on estimates of topology, branch lengths, and bootstrap support. *Syst Biol.* 50:67–86.

Caterino MS, Reed RD, Kuo MM, Sperling FA. 2001. A partitioned likelihood analysis of swallowtail butterfly phylogeny (Lepidoptera: Papilionidae). *Syst Biol.* 50:106–127.

Cognato AI, Vogler AP. 2001. Exploring data interaction and nucleotide alignment in a multiple gene analysis of Ips (Coleoptera: Scolytinae). *Syst Biol.* 50:758–780.

Delsuc F, Stanhope MJ, Douzery EJ. 2003. Molecular systematics of armadillos (Xenarthra, Dasypodidae): contribution of maximum likelihood and Bayesian analyses of mitochondrial and nuclear genes. *Mol Phylogenet Evol.* 28:261–275.

Drummond AJ, Rambaut A. 2007. BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evol Biol.* 7:214.

Ekrem T, Willassen E, Stur E. 2010. Phylogenetic utility of five genes for dipteran phylogeny: a test case in the Chironomidae leads to generic synonymies. *Mol Phylogenet Evol.* 57:561–571.

Elias M, Joron M, Willmott K, et al. (11 co-authors). 2009. Out of the Andes: patterns of diversification in clearwing butterflies. *Mol Ecol.* 18:1716–1729.

Fishbein M, Hibsch-Jetter C, Soltis DE, Hufford L. 2001. Phylogeny of Saxifragales (angiosperms, eudicots): analysis of a rapid, ancient radiation. *Syst Biol.* 50:817–847.

Guindon S, Dufayard JF, Lefort V, Anisimova M, Hordijk W, Gascuel O. 2010. New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. *Syst Biol.* 59:307–321.

Ho SY, Lanfear R. 2010. Improved characterisation of among-lineage rate variation in cetacean mitogenomes using codon-partitioned relaxed clocks. *Mitochondrial DNA* 21:138–146.

Huchon D, Madsen O, Sibbald MJ, Ament K, Stanhope MJ, Catzeflis F, de Jong WW, Douzery EJ. 2002. Rodent phylogeny and a timescale for the evolution of Glires: evidence from an extensive taxon sampling using three nuclear genes. *Mol Biol Evol.* 19:1053–1065.

Jayaswal V, Ababneh F, Jermiin LS, Robinson J. Reducing model complexity of the general Markov model of evolution. *Mol Biol Evol.* Advance Access published May 18, 2011, doi:10.1093/molbev/msr128

Keane TM, Creevey CJ, Pentony MM, Naughton TJ, McInerney JO. 2006. Assessment of methods for amino acid matrix selection and their use on empirical data shows that ad hoc assumptions for choice of matrix are not justified. *BMC Evol Biol.* 6:29.

Le SQ, Lartillot N, Gascuel O. 2008. Phylogenetic mixture models for proteins. *Philos Trans R Soc Lond B Biol Sci.* 363:3965–3976.

Li C, Lu G, Orti G. 2008. Optimal data partitioning and a test case for ray-finned fishes (Actinopterygii) based on ten nuclear loci. *Syst Biol.* 57:519–539.

McGuire JA, Witt CC, Altshuler DL, Remsen JV Jr. 2007. Phylogenetic systematics and biogeography of hummingbirds: Bayesian and maximum likelihood analyses of partitioned data and selection of an appropriate partitioning strategy. *Syst Biol.* 56:837–856.

Miralles A, Kohler J, Vieites DR, Glaw F, Vences M. 2011. Hypotheses on rostral shield evolution in fossorial lizards derived from the phylogenetic position of a new species of Paracontias (Squamata, Scincidae). *Org Divers Evol.* 11:135–150.

Mitchell A, Mitter C, Regier JC. 2000. More taxa or more characters revisited: combining data from nuclear protein-encoding genes for phylogenetic analyses of Noctuoidea (Insecta: Lepidoptera). *Syst Biol.* 49:202–224.

Nguyen MA, Klaere S, von Haeseler A. 2011. MISFITS: evaluating the goodness of fit between a phylogenetic model and an alignment. *Mol Biol Evol.* 28:143–152.

Nylander JA, Ronquist F, Huelsenbeck JP, Nieves-Aldrey JL. 2004. Bayesian phylogenetic analysis of combined data. *Syst Biol.* 53:47–67.

Pagel M, Meade A. 2004. A phylogenetic mixture model for detecting pattern-heterogeneity in gene sequence or character-state data. *Syst Biol.* 53:571–581.

Posada D. 2008. jModelTest: phylogenetic model averaging. *Mol Biol Evol.* 25:1253–1256.

Posada D, Crandall KA. 2001. Selecting the best-fit model of nucleotide substitution. *Syst Biol.* 50:580–601.

Ripplinger J, Sullivan J. 2010. Assessment of substitution model adequacy using frequentist and Bayesian methods. *Mol Biol Evol.* 27:2790–2803.

Ronquist F, Huelsenbeck JP. 2003. MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics* 19:1572–1574.

Shapiro B, Rambaut A, Drummond AJ. 2006. Choosing appropriate substitution models for the phylogenetic analysis of protein-coding sequences. *Mol Biol Evol.* 23:7–9.

Simon C, Buckley TR, Frati F, Steward JB, Beckenbach AT. 2006. Incorporating molecular evolution into phylogenetic analysis, and a new compilation of conserved polymerase chain reaction primers for animal mitochondrial DNA. *Annu Rev Ecol Syst.* 37:545–579.

Stamatakis A. 2006. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics* 22:2688–2690.

Sullivan J, Joyce P. 2005. Model selection in phylogenetics. *Annu Rev Ecol Syst.* 36:445–466.

Telford MJ, Copley RR. 2011. Improving animal phylogenies with genomic data. *Trends Genet.* 27:186–195.