Running head: R package ratematrix

ratematrix: an R package for studying evolutionary integration among several traits on phylogenetic trees

CAETANO, D.S.¹ AND HARMON, L.J.¹

¹Department of Biological Sciences, Institute for Bioinformatics and Evolutionary Studies (IBEST), University of Idaho, Moscow, Idaho, 83843, USA;

Corresponding author: Daniel S. Caetano, E-mail: caetanods1@gmail.com.

Summary

1. Evolutionary integration occurs when two or more phenotypes evolve in a correlated fashion. Correlated evolution among traits can happen due to genetic constraints, ontogeny, and selection and have an important impact on the trajectory of phenotypic evolution. Phylogenetic trees can be used to study such pattern on macroevolutionary time scales by estimating the strength of evolutionary covariance among traits through time and across clades. However, only few applications implement models to conduct comparative analyses of evolutionary integration.

2. We introduce a Bayesian Markov chain Monte Carlo approach to estimate the evolutionary correlation among two or more traits using the evolutionary rate matrix (R). R is a covariance matrix that represents both the rates of evolution of each trait and the structure of evolutionary correlation among traits.

3. Here we present the R package ratematrix, a resource to test hypotheses of evolutionary integration using multivariate data and phylogenetic trees. ratematrix provides a flexible framework allowing for any number of evolutionary rate matrix regimes fitted to the same phylogenetic tree and it incorporates the uncertainty associated with parameter estimates, ancestral state reconstruction and phylogenetic estimation in the analyses.

4. The ratematrix package uses a novel pruning algorithm that significantly improve computational time. We also provide specific functions that facilitate users to conduct long MCMC analysis when computational resources are limited.

Keywords: Anolis, comparative methods, evolutionary integration, evolutionary rates, modularity

This article has been accepted for publication and undergone full peer review but has not been through the copyediting, typesetting, pagination and proofreading process, which may lead to differences between this version and the Version of Record. Please cite this article as doi: 10.1111/2041-210X.12826

This article is protected by copyright. All rights reserved.
Evolutionary changes in one trait are often associated with changes in other traits, such that species traits often do not vary independently of each other (Olson & Miller, 1958). This pattern can be observed in the covariation among traits both within and among populations (Arnold, 1992; Arnold et al., 2001; Revell & Harmon, 2008; Revell & Collar, 2009). The pattern of correlated evolutionary changes among two or more traits is known as evolutionary integration and can be a result of genetic constraints (e.g., pleiotropy), ontogenetic integration, or correlated selection (Arnold, 1992; Arnold et al., 2001; Hansen & Houle, 2004; Goswami et al., 2015; Melo et al., 2016). Although evolutionary integration is ubiquitous across the tree of life, only few comparative methods and associated software applications to date implement models that can estimate evolutionary correlations among traits using phylogenetic trees (Revell & Harmon, 2008; Hohenlohe & Arnold, 2008; Revell & Collar, 2009; Bartoszek et al., 2012; Adams & Otárola-Castillo, 2013; Clavel et al., 2015; Goolsby et al., 2017).

Here we describe the R package ratematrix, which implements a Bayesian estimate of evolutionary rate matrices (\textbf{R}; Revell & Harmon, 2008) fitted to phylogenetic trees and trait data using Markov chain Monte Carlo (as described in Caetano & Harmon, 2017). The \textbf{R} matrix is a variance-covariance matrix that describes the rates of trait evolution under Brownian motion in the diagonals and the evolutionary covariance among traits (i.e., the pattern of evolutionary integration) in the off-diagonals (Revell & Harmon, 2008; Revell & Collar, 2009; Adams & Felice, 2014). With such a matrix we are able to simultaneously investigate the pace of evolution and the structure of evolutionary integration among two or more continuous traits evolving along the branches of a phylogenetic tree. We can also fit multiple \textbf{R} matrices to the same tree in order to test hypothesis of shifts in the evolutionary integration of these traits across clades on the tree.

The \textbf{R} matrix can be estimated using current R packages, however all available implementations rely on point estimates using maximum likelihood. In contrast, the use of a Bayesian framework, as presented here, allows for direct incorporation of uncertainty in parameter estimates in the form of a posterior distribution (Caetano & Harmon, 2017). This is especially important because covariances can be hard to estimate when the number of observations is small relative to the number of parameters in the model, which is commonplace among phylogenetic comparative studies in general.
The model and MCMC implementation

To study the pattern of correlated evolution among two or more continuous traits we use the model described by Revell & Harmon (2008), which consists of a multivariate Brownian motion model with rate equal to the $R$ matrix and root value equal to the vector $a$ (see Equations 2 and 3 in Revell & Collar, 2009). Our implementation allows for multiple independent rate regimes fitted to different branches of the phylogenetic tree (as in Revell & Collar, 2009). Rate regimes can be either fixed \textit{a priori} or a collection of multiple regime configurations can be included in the analysis. For example, multiple regimes applied to the same analysis could be samples from a stochastic character mapping (Huelsenbeck \textit{et al.}, 2003), alternative reconstructions due to missing data, or other plausible hypotheses. However, all regimes need to share the same data at the tips of the tree and same number of rate matrices fitted to the tree. The ratematrix package implements Metropolis-Hastings Markov chain Monte Carlo (MCMC) to estimate the posterior distribution of each $R$ matrix fitted to the tree and the vector of phylogenetic root values ($a$).

Here we detail the proposal distribution used for each set of parameters as well as the options of prior densities currently implemented in the package. At each step of the MCMC chain we choose between the vector of root values and the set of one or more $R$ matrices by drawing from a binomial distribution. The probability that each of these two sets of parameters will be updated is fixed throughout the chain, but can be determined by the user (see function `ratematrixMCMC`). Every time that the set of rate matrices is chosen, only one $R$ matrix is updated but all $R$ matrices fitted to the tree are equally likely to be updated. In contrast, once chosen, the root value for every trait is updated simultaneously. Updates are performed with different configurations of sliding window proposal distributions.

We implemented a uniform distribution with width controlled by the parameter $w_{\mu}$ as the proposal distribution for each element of the vector of phylogenetic means. In contrast, the proposal of $R$ matrices requires a more elaborate scheme, since variance-covariance matrices are constrained to be positive definite. Furthermore, $R$ matrices describe both the rate of evolution of the traits and their pattern of evolutionary correlation, so the proposal distribution needs to provide good mixing for both the variance of each trait and the correlation structure of
such matrices. We implemented a separation strategy (Barnard et al., 2000; Zhang et al., 2006; Liu et al., 2016) proposal scheme which consists of making updates to the vector of standard deviations and the correlation matrix derived from the variance-covariance matrix in separate steps (Figure 1). The vector of standard deviations can be updated directly using a sliding window proposal distribution. On the other hand, the proposal scheme for the correlation matrix requires two steps: first, draw covariance matrices from an inverse-Wishart distribution; and second, we derive correlation matrices from this sample. Finally, we recompose the evolutionary rate matrix in order to calculate the likelihood of the model. Figure 1 shows a diagram that describes the procedure. Note that the vector of standard deviations generated by decomposing the variance-covariance matrix is not evaluated by the likelihood of the model (Zhang et al., 2006). As a result of this parameter-extension approach, we need to correct the acceptance ratio for the transformation from variance-covariance matrix to the corresponding correlation matrix and vector of variances (Figure 1). This proposal scheme allows for independent priors and proposals for the rates of evolution and the evolutionary integration among traits. The ratematrix package allows control over the width of the uniform sliding window proposal for the vector of standard deviations ('w_sd') as well as the degrees of freedom of the inverse-Wishart ('v') used to sample correlation matrices.

The prior densities for the model naturally follow the proposal scheme implemented. Prior densities are determined for the vector of root values, the vector of standard deviations and the correlation matrix (which is sampled by a transformation from the inverse-Wishart distribution). The separation between standard deviations and correlation matrix enable users to translate their biological intuition about the pace and mode of evolution into model parameters in a straightforward manner. The user can set independent priors (options are uniform, normal or log-normal) for both the vector of root values and the vector of standard deviations of the R matrices. For the correlation matrix that, together with the standard deviation, will constitute the R matrix the user can set the degrees of freedom (v) and the scale matrix (Ψ) of the inverse-Wishart distribution. Small values for v make the distribution wider (Figure 2, top) whereas larger values reduce the variance of the distribution, so samples will be closer to Ψ and, as a result, the prior will be more informative (Figure 2, bottom). Note from Figure 2 that a change in the parameters of the inverse-Wishart prior on the correlation matrix will
not change the prior distribution of variances. Since the inverse-Wishart will only be used to sample correlation matrices, Ψ can be set as any correlation matrix or variance-covariance matrix.

**Description of the ratematrix R package**

The package `ratematrix` offers a plethora of functions to allow flexible choices of prior distributions for all parameters in the model, customizable MCMC chains, plots, and robust analyses of convergence (Table 1). The package can be installed from our [github](http://github) repository using the R package `devtools`:

```r
devtools::install_github("Caetanods/ratematrix", build_vignettes = TRUE)
library(ratematrix)
```

The option `build_vignettes` will make the package vignettes available after installation. A list of vignettes can be accessed using:

```r
browseVignettes("ratematrix")
```

We will use the same data from Caetano & Harmon (2017), made available by Mahler et al. (2013) and Moreno-Arias & Calderón-Espinosa (2016), on mainland and island anole lizards as a demonstration of the package. In this study we test whether the radiation of anole lizards from Central and South America to the Caribbean islands was associated with a shift in the pattern of evolutionary integration among morphological traits (head length, tail length and snout-vent length). Here the phylogenetic tree (object `anoles$phy`) is a stochastic map with two regimes produced with the package `phytools`, one rate regime for island and other for mainland species (see Figure Fig S1). Both the trait data and phylogenetic tree are included in the `ratematrix` package.

```r
data(anoles) # Load trait data and phylogeny.
```

**Estimating rates of correlated evolution**

After loading the package and data, we choose the prior distributions for the MCMC chain. We set a uniform prior for the vector of root values and variances and a marginally uniform prior for the covariance matrices, following Barnard et al. (2000) (see also documentation for `makePrior`). The marginally uniform prior produces uniform distributions for each of the covariance terms.
(\(\sigma^2_{i,j}\) for \(i \neq j\)) of the variance-covariance matrix after integrating over the uncertainty of the other parameters (i.e., the marginal distribution for \(\sigma^2_{i,j}\)). Many characteristics of the Markov chain Monte Carlo can be customized (see documentation for `ratematrixMCMC`). We encourage users to run short preliminary chains in order to adjust the width of the proposal distributions for each set of parameters in function of the acceptance ratio (see function `logAnalyzer`) prior to a full MCMC chain analysis. This procedure can improve the mixing of the chains, which might decrease the number of generations required until convergence and increase the effective sample size (ESS) of the posterior distribution. The following lines of code will run only a short example, starting with a random sample from the prior distribution. The package also provide results from previous MCMC analyses with this same data as examples.

```r
estimateTimeMCMC(data=anoles$data[,1:3], phy=anoles$phy, gen=10000)
handle <- ratematrixMCMC(data=anoles$data[,1:3], phy=anoles$phy,
, prior="uniform", gen=10000)
```

The `estimateTimeMCMC` function estimates the time for the MCMC chain whereas `ratematrixMCMC` runs it. The MCMC function writes one file with the parameter samples and another with the log information for each generation. Both files are marked with a unique identifier that prevents multiple chains of overwriting each other. The `handle` object is a list containing detailed information about the MCMC chain and is required in order to read the posterior distribution from files, analyze the log information and continue an unfinished MCMC chain. Below we show an example of how to read the posterior distribution from the files. Then, we make plots (Figure 3) and calculate summary statistics based on results of a converged MCMC chain provided as example data.

```r
  (short_chain <- readMCMC(handle, burn=0.25, thin=1) )
logAnalyzer(handle, burn=0.25, thin=1) # Log information for the chain.
plotRatematrix(chain=short_chain) # Plots the short chain.
plotRootValue(chain=short_chain) # Plots the short chain.
```

```r
data(anolesPost) # Load example of posterior distribution.
plotRatematrix(chain=anolesPost$chain1)
plotRootValue(chain=anolesPost$chain1)
```

This article is protected by copyright. All rights reserved.
checkConvergence(anolesPost$chain1, anolesPost$chain2)

testRatematrix(chain=anolesPost$chain1, par="correlation")

testRatematrix(chain=anolesPost$chain1, par="rates")

The logAnalyzer function calculates the acceptance ratio for each parameter of the model and plots the trace of the log-likelihood for the MCMC chain. The plot functions show the posterior distribution of parameter estimates. plotRatematrix produces a plate with histograms for the rate of evolution of each trait (diagonal) and the pairwise evolutionary covariation among traits (upper-diagonal). The lower-diagonal plots show ellipses for the 95% confidence interval of the bivariate distribution between each pair of traits. Different from the histograms, the ellipses are only a sample from the posterior distribution (see documentation for plotRatematrix).

Here we applied the test of convergence for the chains using Gelman & Rubin (1992) potential scale factor analysis (see function checkConvergence). This convergence test requires two or more independent MCMC chains and compares the variance of parameter estimates between chains and within each chain.

The testRatematrix function calculates a series of summary statistics based on the pairwise degree of overlap among the posterior distribution of R matrices fitted to same phylogenetic tree (Caetano & Harmon, 2017). If this overlap exceeds 5%, then we can conclude that the difference between posterior parameter estimates is not strong enough to support the hypothesis that regimes are representations of distinct macroevolutionary patterns. When we compare the posterior distribution for the evolutionary correlation (par="correlation"; overlap of 0.4) and the rates of evolution for each trait (par="rates"; overlap of 0.0002) there is no evidence for a shift in the pattern of evolutionary integration but island anole lineages show faster rates of trait evolution when compared to mainland lineages (Figure 3, see also Caetano & Harmon, 2017). We refer readers to Caetano & Harmon (2017) for an extensive simulation study of the performance of the method as well as the use of summary statistics under a diverse set of scenarios of correlated evolution.

Integration of uncertainty in regime configurations

One of the advantages of Bayesian implementations is that analyses can integrate uncertainty from different sources. Rate regimes, for example, are used to map the set of nodes and branch

This article is protected by copyright. All rights reserved.
lengths that will be assigned to each $R$ matrix fitted to a phylogenetic tree. Such regimes are usually determined \textit{a priori} by ancestral estimate, since we often are interested in the association between some characteristic and a possible shift in the pattern of evolutionary integration among traits. However, ancestral state estimates can be uncertain and alternative reconstructions are often possible, specially when the states of the characteristics under study are polymorphic or of dubious interpretation. Nevertheless, most comparative methods are implemented to estimate the parameters of the model with a single regime configuration and users need to perform multiple independent analyses in order to incorporate uncertainty associated with ancestral state estimates.

The package \texttt{ratematrix} offers a different approach by allowing a pool of phylogenetic trees to be directly incorporated in the MCMC. This pool can comprise repeated simulations from a stochastic mapping analysis, equally parsimonious ancestral state reconstructions or even a random sample of trees from the posterior distribution of a phylogenetic inference analysis. In order to incorporate the pool of trees, we randomly sample one phylogenetic tree each time the likelihood function of the model is evaluated. Note that this procedure is not the same as a joint Bayesian MCMC estimate of the trait model and the phylogenetic tree because the posterior distribution of rate regimes or phylogenetic trees is not sampled as part of the MCMC. However, the posterior distribution of parameter estimates for the model is sampled conditioned on the pool of trees provided by the user. Bellow we show how to create an analysis based on a pool of stochastic mapping simulations:

```r
library(phytools) # Using Revell (2012).
state <- setNames(anoles$data$Location, rownames(anoles$data))
phy_map <- make.simmap(anoles$phy, x=state, nsim=100)
handle_map <- ratematrixMCMC(data=anoles$data[,1:3], phy=phy_map , prior="uniform", gen=10000, outname="phy.pool")
logAnalyzer(handle_map, burn=0.25, thin=1)
```

When a pool of phylogenetic trees is provided for the MCMC, \texttt{logAnalyzer} returns the acceptance ratio for each of the phylogenetic trees. Relative low acceptance ratio for a given tree means that proposals were rejected more often than with the rest of trees from the same pool. This might be due to a low likelihood score for the multivariate Brownian-motion model
given the tree as a result of an unlikely topology, branch lengths or rate regime configuration. If this is the case, one can check if the tree has some particular attributes that make it distinct from other trees in the pool, since such patterns might carry important biological information. Furthermore, we recommend that an independent analysis is performed with such a tree (or trees) so that one can test whether results are significantly different than the former analysis using the entire pool.

**Continuing unfinished chains or adding extra iterations**

The ratematrix package allows for continuing an unfinished MCMC analysis or to append additional generations to the previous MCMC chain. This is an essential feature given the computational burden associated with any Bayesian simulation approach. In both cases, the user needs to provide the handle object returned by the ratematrixMCMC function (or saved to the working directory). The following example will add iterations to the previous MCMC chain:

```r
handle_map_add <- continueMCMC(handle_map, add.gen=1000)
```

One can use continueMCMC alongside checkConvergence to add generations to the MCMC until the chain(s) pass the convergence test. This is especially relevant given that the number of generations required for acceptable convergence is dependent on the data and the configuration of the sampler.

**New pruning algorithm improves computational time**

The package ratematrix implements a novel algorithm to evaluate the likelihood function of a multivariate Brownian motion model when two or more R matrix regimes are fitted to the same phylogenetic tree (Caetano & Harmon, 2017). In previous implementations, a large matrix composed by the multiplication between the phylogenetic variance-covariance matrix, with dimension equal to the number of species in the tree, and the evolutionary rate matrix, with dimension equal to the number of traits, needed to be computed. However, any operation with such large matrices can become very computationally intensive. Recently, Caetano & Harmon (2017) implemented an extension of Felsenstein (1973) pruning algorithm that avoids such calculations. As a result, matrix operations need only to be performed with the evolutionary
rate matrix \((R)\), which is usually a fairly small matrix. Figure 4 shows the computational time for the likelihood function under different approaches. Computation using the full inverse and determinant of the matrices is the approach that scales worst with number of traits and size of the phylogeny. Although the ‘rpf’ method (Gustavson et al., 2010), which avoids the computation of the full inverse and determinants, shows a significant improvement, the pruning algorithm has the best performance. With respect to the asymptotic upper bounds \((O)\), all approaches scale equally with the number of traits in the analysis when the size of the phylogeny is held constant, but there is a remarkable improvement with the scaling in function of the number of tips in the phylogeny. The pruning algorithm scales with \(O(n + r^3)\) whereas the other methods scale with \(O(n^3 + r^3)\), where \(n\) is the number of tips in the phylogeny and \(r\) is the number of traits. The reduction of time to evaluate the likelihood of the model is fundamental to the implementation of simulation based approaches such as the Bayesian Markov chain Monte Carlo estimates performed by the package ratematrix (Caetano & Harmon, 2017).

**Resources**

ratematrix is an open-source R package that can be installed from the github repository https://github.com/Caetanods/ratematrix. A series of tutorials are available at https://github.com/Caetanods/ratematrix/wiki.

**Data accessibility**

Data used in this study were compiled from Mahler et al. (2013) and Moreno-Arias & Calderón-Espinosa (2016) and is available at https://doi.org/10.6084/m9.figshare.5037839.v1.

**Authors’ contributions**

D.S.C. conceived the methodology, implemented the software and wrote the manuscript, package documentation, and tutorials. L.J.H. contributed to conceive the methodology, wrote the manuscript and tested the software. All authors gave final approval for publication.

This article is protected by copyright. All rights reserved.
Acknowledgements

D.S.C. was supported by a fellowship from Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES: 1093/12-6) and from Bioinformatics and Computational Biology Program at the University of Idaho in partnership with IBEST (the Institute for Bioinformatics and Evolutionary Studies). L.J.H. was supported by a grant from National Science Foundation (award DEB-1208912). We thanks Julien Clavel and an anonymous reviewer for suggestions that improved the quality of the manuscript.
References


This article is protected by copyright. All rights reserved.


This article is protected by copyright. All rights reserved.


**Supporting Information**

Details of electronic Supporting Information are provided below.

**Figure S1.** Maximum clade credibility phylogenetic tree for anole lizards with branches painted to show mainland and island lineages.
Table 1: Principal functions available in *ratematrix*.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>checkConvergence</td>
<td>Perform tests of convergence with one or multiple MCMC chains.</td>
</tr>
<tr>
<td>continueMCMC</td>
<td>Continue or add generations to a MCMC chain.</td>
</tr>
<tr>
<td>estimateTimeMCMC</td>
<td>Estimate the time that a MCMC chain will take to run.</td>
</tr>
<tr>
<td>likelihoodFunction</td>
<td>Compute the log-likelihood of the multivariate Brownian-motion model.</td>
</tr>
<tr>
<td>logAnalyzer</td>
<td>Compute acceptance ratio for parameters and pool of phylogenies and make trace plots for the log-likelihood and acceptance ratio.</td>
</tr>
<tr>
<td>makePrior</td>
<td>Create prior densities for the model.</td>
</tr>
<tr>
<td>makeStart</td>
<td>Create starting point for the MCMC chain.</td>
</tr>
<tr>
<td>mergePosterior</td>
<td>Merge multiple chains from the same data into a single chain.</td>
</tr>
<tr>
<td>mergeSimmap</td>
<td>Merge rate regimes mapped to a phylogenetic tree.</td>
</tr>
<tr>
<td>plotPrior</td>
<td>Plot the prior distribution for the parameters of the model.</td>
</tr>
<tr>
<td>plotRatematrix</td>
<td>Plot the posterior distribution of evolutionary rate matrices (R).</td>
</tr>
<tr>
<td>plotRootValue</td>
<td>Plot the posterior distribution of root values (phylogenetic mean).</td>
</tr>
<tr>
<td>ratematrixMCMC</td>
<td>Make the Bayesian Markov chain Monte Carlo analysis.</td>
</tr>
<tr>
<td>readMCMC</td>
<td>Read MCMC samples from the <em>ratematrixMCMC</em> output file.</td>
</tr>
<tr>
<td>samplePrior</td>
<td>Draw samples from the prior density created by <em>makePrior</em>.</td>
</tr>
<tr>
<td>simRatematrix</td>
<td>Simulate data given a phylogenetic tree, evolutionary rate matrix and root values.</td>
</tr>
<tr>
<td>testRatematrix</td>
<td>Use summary statistics to test for shifts between rate regimes.</td>
</tr>
</tbody>
</table>
Proposal strategy for R matrices

* or any distribution of positive values

covariance
matrix

inv-Wishart
distribution

transformation

T

variance
vector

correlation
matrix

R

\[ J_T = \prod_{i=1}^{q} d_i^{q-1} \]

Figure 1: Diagram of the separation strategy proposal (Barnard et al., 2000). Boxes in grey show the proposal distributions for the variance vector and correlation matrix that compose the evolutionary rate matrix (R). Boxes in blue show the elements that are directly (or indirectly, in the case of the covariance matrix) evaluated in the acceptance step of the MCMC. The yellow circle shows the transformation (T) required to decompose the variance-covariance matrix sampled from an inverse-Wishart into a correlation matrix and the variance vector. The yellow square shows the formula for the Jacobian correction due to T (Zhang et al., 2006), where \( d_i \) stands for the variance of traits 1 to q. The red square demonstrates that an additional variance vector is produced in the process, but it is discarded. The green circle is a representation of the R matrix as a product between the variance vector and the correlation matrix.
Figure 2: Samples from the prior of the evolutionary rate matrix ($R$) for two simulated traits using the separation strategy (Barnard et al., 2000). Standard deviation was modelled as a uniform distribution between 0 and 10 and the correlation matrix was derived from a inverse-Wishart centered on a scale matrix with positive correlation ($\text{corr} = 0.5$). Top figure shows a weak prior with small value for the degrees of freedom parameter ($\nu = 3$) whereas bottom figure are draws from a more informative prior ($\nu = 12$). In each figure, the plots in the diagonal show evolutionary rates for each trait whereas the upper-diagonal plot shows the evolutionary covariation. Lower-diagonal plot show 150 randomly sampled ellipses representing the 95% quantile for the bivariate distribution. Although both priors share the same scale matrix, when $\nu$ is small the prior distribution has more variance than when $\nu$ is larger. The diagonal plots are held constant since the prior distribution for the standard deviation is the same in both figures. Note that ellipses show both positive and negative correlation when the prior is weak (top figure) and positive or no correlation when the prior is more informative (bottom figure).
Figure 3: Posterior distribution of the evolutionary rate matrix ($R$) regimes fitted to the island anole (gray) and mainland anole (pink) lineages. A different $R$ matrix were jointly estimated for each regime. The plots in the diagonal show evolutionary rates (variances) for each trait; $\sigma_{1}^{2}$ for SVL, $\sigma_{2}^{2}$ for tail length, and $\sigma_{3}^{2}$ for head length. Upper-diagonal plots show pairwise evolutionary covariance (covariances); $\sigma_{1,2}^{2}$ between SVL and tail length, $\sigma_{1,3}^{2}$ between SVL and head length, and $\sigma_{2,3}^{2}$ between tail length and head length. The ellipses in the lower-diagonal plots represent the 95% confidence interval of each bivariate distribution for 50 randomly sampled $R$ matrices from the posterior. The order of the ellipse plots is a mirror reflection from the upper-diagonal evolutionary covariance plots. Ellipses are only a sample of the posterior because a very large number of lines can become hard to visualize, however the user can set any number of samples (or the entire posterior).
Figure 4: Time in seconds spent to compute the likelihood function using different approaches. Top figure shows computational time for two traits and a phylogeny of different number of species. Bottom figure shows computational time with a phylogeny of 400 species and increasing number of traits. Both plots show a comparison among three approaches: ‘inverse’ uses the full inverse and determinant of matrices as implemented in phytools (Revell, 2012), ‘rpf’ uses the rectangular full-packed format algorithm as implemented in mvMORPH (Clavel et al., 2015), and ‘pruning’ uses Felsenstein (1973) pruning algorithm as implemented in ratematrix.