Chapter 4
Graphical Methods for Visualizing Comparative Data on Phylogenies

Liam J. Revell

Abstract Phylogenies have emerged as central in evolutionary biology over the past three decades or more, and an extraordinary expansion in the breadth and sophistication of phylogenetic comparative methods has played a large role in this growth. In this chapter, I focus on a somewhat neglected area: the use of graphical methods to simultaneously represent comparative data and trees. As this research area is theoretically very broad, I have concentrated on new methods developed by me, or techniques devised by others and implemented by me as part of my R phylogenetics package, phytools. I describe a variety of methods in this chapter, including approaches that can be used to map reconstructed discrete or continuous character evolution on trees; techniques for projecting phylogenetic trees into morphospace; and methods for visualizing phylogenies in the context of a global or regional geographic map. In this chapter, my intention is not merely to showcase new methods that I have developed. Rather, I have also dedicated considerable attention to detailing the algorithms and computational techniques required for these approaches with the hope that this chapter will become a resource or jumping-off point for researchers interested in building new, more advanced approaches and methods in this area.

4.1 Introduction

No one would seriously dispute the contention that a well-designed and informative figure can replace at least a thousand words, if not more, in a contemporary scientific publication. Visualization can also play an integral role in the preliminary analysis of new data and in generating new hypotheses which can be

L. J. Revell (✉)
Department of Biology, University of Massachusetts Boston, Boston, MA 02125, USA
e-mail: liam.revell@umb.edu

L. Z. Garamszegi (ed.), *Modern Phylogenetic Comparative Methods and Their Application in Evolutionary Biology*, DOI: 10.1007/978-3-662-43550-2_4,

explored with more rigorous tests. However, effective graphical methods for new types of data and analysis in phylogenetic comparative biology may also require some creative new method development for visualization (e.g., Sidlauskas 2008; Revell 2013). In this chapter, I'm going to describe and illustrate several new approaches—devised by myself or others and implemented in my R package, phytools (Revell 2012)—for visualizing comparative data on phylogenies. Specifically, I'll focus on visualization methods that can simultaneously show the phylogenetic tree and a set of comparative data for discrete and continuously valued phenotypic traits.

Large evolutionary changes take place over thousands of generations to millions of years. In many cases, phylogenetic comparative biology—the theory and practice of drawing evolutionary inferences from phylogenies and comparative data for phenotypic characters—represents our best or only recourse for studying evolution on these vast timescales (Felsenstein 1985, 1988; Harvey and Pagel 1991; Mahler et al. 2010; Nunn 2011). Phylogenetic comparative methods have advanced considerably in recent years (e.g., Butler and King 2004; O'Meara et al. 2006; Bokma 2008; Fitzjohn 2010; Eastman et al. 2011; Felsenstein 2012; Revell et al. 2012; Beaulieu et al. 2013; Revell 2014; reviewed in Glor 2010; O'Meara 2012; Pennell and Harmon 2013). Many of the chapters of this book exemplify these great strides. However, in some cases, these new methods and new types of data for comparative biology also present us with new challenges in visualization. Specifically, the most efficient, visually appealing, and informative way to simultaneously represent phylogenetic and phenotypic information in a single plot is not always clear.

Since a simple plot of the phylogeny forms the basis for several of the visualization methods that I'll describe in this chapter, I'm going to begin (in Sect. 4.2, below) by detailing the general algorithm that can be used to draw two common types of tree plots. In subsequent sections, I'll focus my attention more specifically on the challenges of simultaneously visualizing phylogenetic relationships and trait data for phenotypic characters. In Sect. 4.3, I'll concentrate on discrete character methods. I'll describe the comparative method called stochastic character mapping (Nielsen 2002; Huelsenbeck et al. 2003; Bollback 2006) and illustrate how a single stochastic map can be plotted on the branches and nodes of a phylogeny (Sect. 4.3.1). Next, I'll detail and illustrate two different approaches for aggregating the results of many stochastic mappings (Sects. 4.3.2 and 4.3.3; Revell 2013). Then, in Sect. 4.4, I'll move on to several different methods that have been developed for continuously valued phenotypic traits. The first and second methods (described in Sects. 4.4.1 and 4.4.2) involve some kind of projection of the phylogeny into a space that is either fully or partially defined by our phenotypic trait data in two or three dimensions (e.g., Sidlauskas 2008; Evans et al. 2009). The third method (described in Sect. 4.4.3) involves directly mapping the reconstructed evolution of a continuous trait onto the branches of a plotted tree. I also show how we can combine both types of plots to create a "phylogenetic scatterplot matrix" suitable for multidimensional continuous trait data. In Sect. 4.5, I'll describe a few additional new approaches, including the projection of a tree onto a geographic map, and the combination of discrete and continuous character methods into a single plot. In Sect. 4.6, I'll give a brief introduction to the technical matter of programming phylogeny plotting methods in R (R Core Team 2013). Finally, in Sect. 4.7, I'll try to provide some concluding thoughts on the challenges of visualizing phylogenetic and comparative data, the "paper paradigm" (Rosindell and Harmon 2012), and some possible future developments of this field.

Before I begin, I should emphasize again that this chapter is not intended as a comprehensive survey of phylogenetic visualization methods, nor even of visualization methods for phylogenetic comparative biology. Rather, I have focused specifically on methods that I have worked on in some way. By providing detail on these methods, rather than a superficial survey of all approaches, I hope to inspire readers of this chapter to think about novel techniques that are suitable for their (idiosyncratic or general) problem or data. Hopefully, the content of this chapter can become a starting point for additional methodological innovation and discovery by other researchers.

Although all the methods of this chapter are implemented in my phytools R package, plotting methods in phytools make extensive use of R base graphics (R Core Team 2013), as well as some other packages such as scatterplot3d (Ligges and Mächler 2003), maps (Becker et al. 2013), plotrix (Lemon 2006), and rgl (Adler and Murdoch 2013). In addition, phytools depends internally on ape (Paradis et al. 2004) and phangorn (Schliep 2011) for their extensive suite of functions for reading, writing, manipulating, and analyzing phylogenetic trees.

4.2 The General Problem of Drawing Trees

In this section, I'll briefly describe the basic general algorithm for taking a tree stored in computer memory (or, hypothetically, in your own memory) and drawing that tree onto a piece of paper or a plotting object in R. Since there are already many different tools for tree drawing available, this section will primarily appeal to researchers interested in programming new visualization methods for phylogenies (or understanding how existing methods are programmed). Readers that are not interested in such things can probably skip this section. Here, I'm going to focus on the general algorithms for tree plotting, which would apply equally to any programming language or development environment; however, in Sect. 4.6, toward the end of this chapter, I'm going to go on to provide some specific code in R to replicate one of these algorithms. Here, I'm going to concentrate on two different types of trees that are also among the easiest to draw: square and circular phylograms with intermediate node placement (Felsenstein 2004). Examples of these two tree-plotting methods are illustrated in Fig. 4.1a and b, respectively. I have to presume that these algorithms have previously been independently discovered by anyone who has ever programmed a tree-plotting function; however,

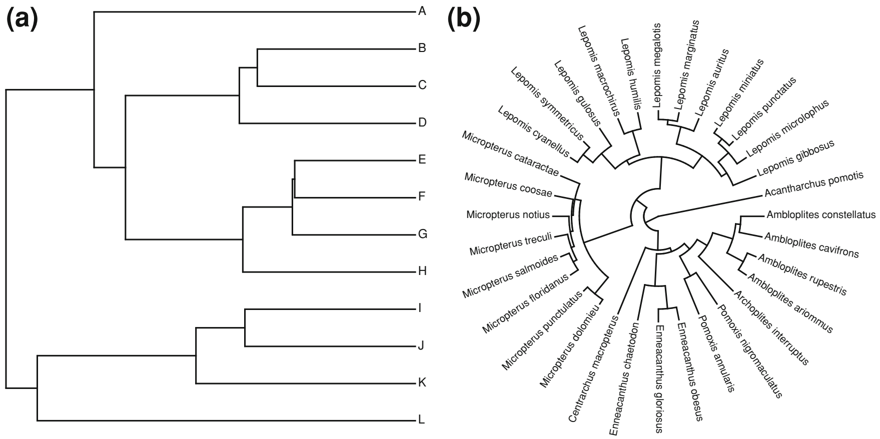

Fig. 4.1 **a** A square phylogram representing a simulated random tree. **b** A circular phylogram showing the empirical phylogeny of Centrarchidae [from Near et al. (2005)]

I'm not aware (and I could be wrong, of course) of them having been written down—at least not in the phylogeny literature.

Figure 4.1a shows a stochastic, pure-birth tree (i.e., a "Yule tree") plotted as a rightward square phylogram with intermediate node placement (Felsenstein 2004). *Rightward* refers to the orientation of edges (when taken as vectors leading from parent to daughter); *phylogram* just means that the plotted edges are proportional in length to the branches of the tree; and *intermediate node placement* refers to the (in this case) vertical position of ancestral nodes—in other words, we have positioned them vertically intermediately between the uppermost and lowermost daughter nodes. Note that although the algorithm is described specifically for a rightward orientation (in other words a phylogeny that "grows" from left to right on our page; Felsenstein 2004), changing to a leftward orientation, or an upward or downward orientation, simply requires that we change the sign of *x*, or flip *x* and *y*, or do first one and then the other.

To create a graph in this style, the first step (step 1) is assigning vertical positions to all of the tips in the tree. To do this, we first have to sort the tips into what I'm going to refer to as *cladewise* order (Paradis 2012). This just means that tips in a clade are adjacent to each other in the ordering. In the case of Fig. 4.1, this means that *A*, *B*, *C*, *D*, *E*, *F*, *G*, *H*, *I*, *J*, *K*, *L*; *B*, *C*, *D*, *E*, *F*, *G*, *H*, *A*, *I*, *J*, *K*, *L*; or *E*, *F*, *G*, *H*, *B*, *C*, *D*, *A*, *I*, *J*, *K*, *L* are all valid cladewise orderings of the twelve taxa in our tree. (There are also many other valid orderings.)

If this ordering seems like it could be complicated to obtain, then it might be helpful to note that a left-to-right (or right-to-left) reading of the tip labels in a Newick style tree is guaranteed to produce tip labels in cladewise order. For example, the Newick strings below:

((A:0.78,(((B:0.39,C:0.39):0.04,D:0.43):0.28,(((E:0.29,F:0.29):0.01,G:0.3):0.12, H:0.42):0.29):0.08):0.22,(((I:0.42,J:0.42):0.12,K:0.54):0.39,L:0.92):0.08);, *(((((B:0.39,C:0.39):0.04,D:0.43):0.28,(((E:0.29,F:0.29):0.01,G:0.3):0.12,H:0.42): 0.29):0.08,A:0.78):0.22,(((I:0.42,J:0.42):0.12,K:0.54):0.39,L:0.92):0.08);*, and *((((((E:0.29,F:0.29):0.01,G:0.3):0.12,H:0.42):0.29,((B:0.39,C:0.39):0.04,D:0.43): 0.28):0.08,A:0.78):0.22,(((I:0.42,J:0.42):0.12,K:0.54):0.39,L:0.92):0.08);*

are all valid representations of the tree in Fig. 4.1a, and thus, all contain tip labels (read left to right or right to left) in a cladewise ordering. Newick tree format is the most widely used way to record phylogenies in plain text. Newick format uses parentheses, commas, and colons to represent hierarchical and sister relationships, and branch lengths, respectively (Archie et al. 1986). For more information about the Newick format, readers should refer to Felsenstein (2004).

Having ordered the tips in this way, we can now go ahead and assign each tip a vertical position evenly spaced from 1 through n for n tips (step 2). This step is shown in Fig. 4.2a. (We could have just as well assigned values from n through 1, $-n/2$ through $n/2$, 100 through $100 \times n$, or 0.1 through $0.1 \times n$, etc., so long as we are prepared to resize the vertical axis of our plotting area accordingly.)

Now, we conduct a post-order traversal of the tree. This means we descend from the tips to the root of the tree passing through each daughter node before its parent. At each node, we can assign a vertical position to the node (and its preceding edge) that is intermediate between the two daughters (step 3). This step is illustrated in Fig. 4.2b, in which the horizontal dashed line indicating the vertical position of each internal node is labeled with a list of the tips descended from that node. If there are more than two daughters, in other words, if the node contains a multifurcation, we compute the average of the lowermost and the uppermost daughters (Felsenstein 2004). Having done this for all internal nodes, we are now in possession of the vertical position of all plotted edges in the tree.

The next step (step 4) is to compute the horizontal starting and ending points of each branch in the tree. To do this, we start at the root and use a pre-order tree traversal, which means that we traverse each parent node before its daughters. As we traverse the tree up from the root, we compute the starting horizontal position of an edge as the sum of all preceding branch lengths in the path from the root to the parent (starting) node of that edge. The ending point of the same edge is merely this value plus the branch length of the current edge. The points computed in this step are shown in Fig. 4.2c. Now, we have the vertical positions (from step 3) and the starting and ending points of each branch in the tree. When we plot the horizontal lines that connect these points, we've plotted all the branches in our phylogeny in their correct horizontal and vertical positions. This step is shown in Fig. 4.2d.

To add the relationships between species and clades (step 5), in other words, the vertical lines in our plot of Fig. 4.1a, we start by taking each internal node in the tree including the root. We go to its height on the horizontal axis, and we draw a vertical line connecting the uppermost and lowermost vertical positions of its two or more daughters. This step is illustrated in Fig. 4.2e. Having ordered the tip taxa

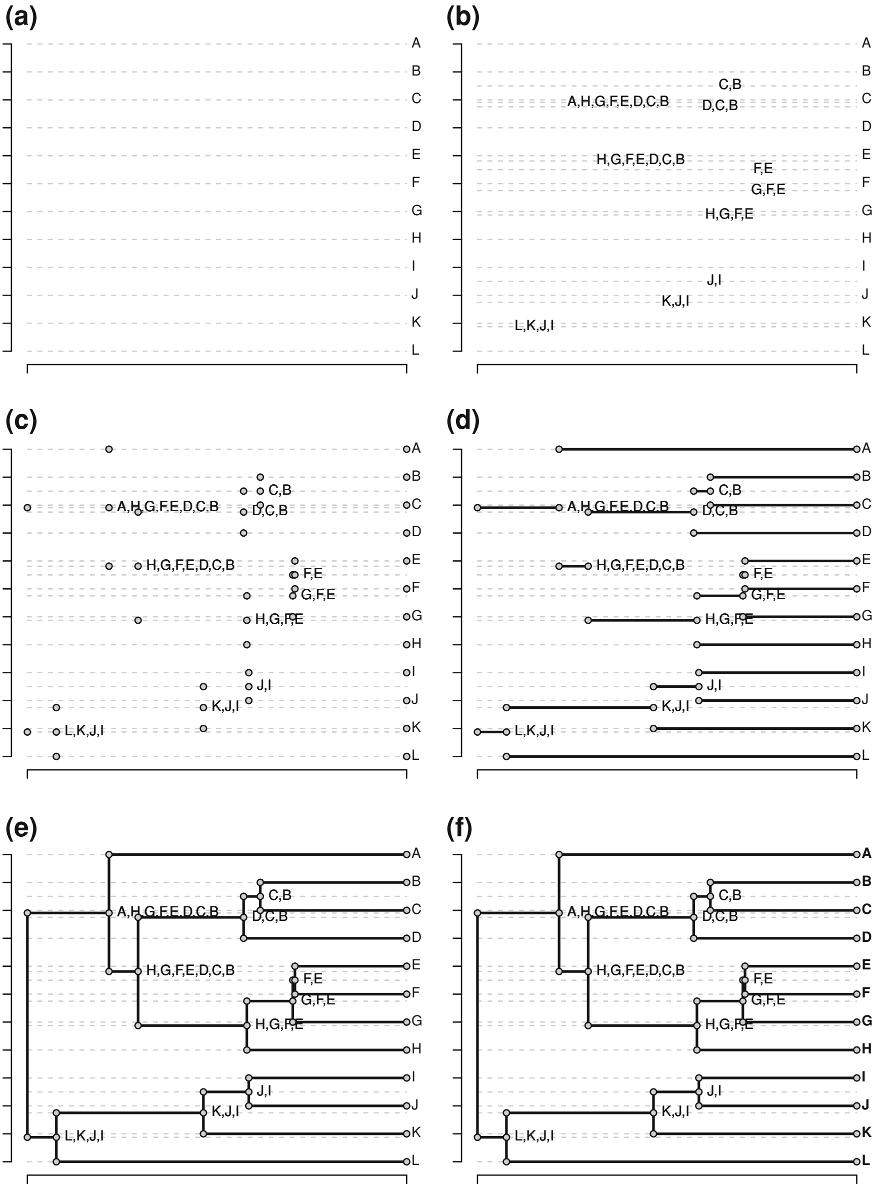

Fig. 4.2 An illustration of the algorithm for drawing a rightward square phylogram with intermediate node placement using the simulated tree of Fig. 4.1a. **a** Order the tips of the tree in "cladewise" order. **b** Conduct a post-order traversal of the tree and compute the vertical position of internal edges as the average of the highest and lowest daughter edges. The vertical positions of terminal and internal edges of the tree are shown as horizontal dashed lines in panels (**a**) and (**b**). **c** Conduct a pre-order tree traversal and record the height above the root of the starting and ending points of each edge. **d** Draw edges. **e** Plot the relationships between edges by going to each internal node and adding a vertical line connecting the highest and lowest daughter edges. **f** Add tip labels at the end of each terminal edge

by clade before assigning their vertical positions in step 2, we've guaranteed that our tree is "untangled"; in other words, no vertical lines cross any of our horizontally drawn edges. Finally, to include labels (step 6), we merely add text to our plot for taxa 1 through n at each of the 1 through n vertical positions we assigned in step 2, in this case using the horizontal positions of the end of each terminal edge, computed in step 4. This final step is illustrated in Fig. 4.2f.

For circular phylograms, such as the phylogeny of centrarchid fishes from Near et al. (2005) shown in Fig. 4.1b, we conduct steps 1 through 4, just as described above for rightward square phylograms. Then, however, we translate our node heights (above the root, two for each edge) and vertical edge positions to the coordinate system of our circular plot using the formulae $\mathbf{x}_i = \mathbf{r}_i \cdot \cos(Y_i)$ and $\mathbf{y}_i = \mathbf{r}_i \cdot \sin(Y_i)$, where $\mathbf{r}_i$ (radius) is the set of heights above the root for edge i and Y_i is its vertical position in the square tree. We connect each parent and daughter subtending edge i using a radial line from $(x_{1,i}, y_{1,i})$ to $(x_{2,i}, y_{2,i})$. Then, at each internal node, j, we draw an arc of radius r_j spanning the lowermost to the uppermost daughter edges of j. Finally, we plot our 1 through n labels at the end of each terminal edge. Normally, we would angle the labels using the same angle as the terminal edge, but then flip the orientation of the label by 180° for labels plotted between 90° and 270° from horizontal (e.g., see Fig. 4.1b). Circular or fan-style plots provide the advantage of allowing larger phylogenies to be represented in the same plotting area (e.g., Edwards et al. 2010), sometimes even with readable tip labels; however, they create the disadvantage that because time since the root is represented by radial distance from the origin (rather than horizontal distance), contemporaneous nodes and tips are a little bit more difficult to identify.

4.3 Discrete Character Methods

4.3.1 Mapping a Single Discrete Character on the Tree

The first plotting method for comparative data that I am going to describe is the relatively simple approach of visualizing the reconstructed history of a discretely valued character trait obtained from a phylogenetic method called stochastic character mapping (Nielsen 2002; Huelsenbeck et al. 2003). Stochastic mapping is a procedure in which we randomly sample possible character histories for a discrete trait such that the probability of sampling any specific history varies in direct relation to its posterior probability under our model of trait evolution (generally, a continuous-time discrete-state Markov chain), given our tree and data. Stochastic character mapping is described in more detail by Nielsen (2002), Huelsenbeck et al. (2003), and Bollback (2006).

Briefly, to generate a single stochastic character map on the tree, we first sample a joint reconstruction of our discrete character across all the nodes of the tree conditioned on an instantaneous transition matrix between states, $\boldsymbol{Q}$, and our discrete character data, **x**. (Where $\boldsymbol{Q}$ comes from we will leave aside for the moment.) These states are sampled from their joint posterior probability distribution following Bollback (2006). Next, we simulate changes along the edges of the tree using a rejection procedure. We obtain the waiting times for changes between states by drawing randomly from an exponential distribution with rate $-Q_{ii}$, given initial state i. If the time is shorter than the total branch length of the current edge, we simulate another change, and then another, and so on, until we reach the end of the branch. At each change, we determine the new state by picking a state at random with probability $\Pr(j) = Q_{ij} \Big/ \sum_{j=1}^{n, j \neq i} Q_{ij}$, for any derived state j. Here, n is the total number of different states for our discrete character. If the starting and ending states for the branch match our stochastic joint sampled node states, we have successfully simulated a stochastic history for that branch. If not, then we reject our simulation and repeat it until we have sampled a history with starting and ending states that agree with our stochastically sampled states for the nodes subtending that branch. Stochastic mapping is implemented in phytools (Revell 2012).

There are two different procedures that we can use to obtain our continuous-time discrete-state Markov chain transition matrix, $\boldsymbol{Q}$. We can sample $\boldsymbol{Q}$ using Bayesian MCMC, which I'll refer to as the full hierarchical Bayesian approach; or we can fix $\boldsymbol{Q}$ at its most likely value, which I'm going to call an empirical Bayesian approach (e.g., Yang 2006). The latter is unbiased, but has the problem that variables (such as the number of transitions between states) that are estimated from a posterior sample of stochastically mapped trees in which $\boldsymbol{Q}$ is set to its most likely value will tend to have variance that is slightly too low. Conversely, parameter estimates that we obtain from the full hierarchical Bayesian approach, in which $\boldsymbol{Q}$ and the stochastic histories are sampled from their joint posterior probability distribution, should generally have the correct variance; however, this approach depends on the somewhat difficult task of specifying a reasonable prior probability density for $\boldsymbol{Q}$.

When we've figured out the best approach for our tree and data, and then generated one or multiple stochastic maps, we can easily plot the stochastic character maps on a tree using different colors to map different character states through time. This is accomplished by computing the fraction of time spent in each state along each plotted edge in the tree. Having done this, we can then plot each state using a different colored line segment. For the lines connecting branches that share a common ancestor (i.e., the vertical or curved lines in Figs. 4.1a and b, respectively), we merely plot this line using the color of the last state on the preceding edge. (We could equally well use the initial state for any daughter edge, since under a continuous-time character evolution model, it is theoretically impossible that the character changes state exactly at a node.) Figure 4.3 shows an example circular tree for Greater Antillean anole species from Mahler et al. (2010) with a mapped discrete character "ecomorph"—the famous convergent ecological and morphological habitat specialists found in the *Anolis* lizard fauna of the Caribbean (Losos 2009). (The pie charts at internal nodes are not part of the stochastic map and will be explained in Sect. 4.3.2, below.) To generate this stochastic map, I assumed a single substitution rate between all pairs of states, and then I fixed this rate to its most likely (i.e., maximum likelihood) value (Pagel 1994), making this an example of empirical Bayesian stochastic mapping (outlined above).

4.3.2 Aggregating Stochastic Maps: Node Posterior Probabilities

One of the difficulties that is inherent in visualizing the results of stochastic mapping using a plot like that of Fig. 4.3 is that plotting only one stochastic character map can create the misleading impression of certainty in the discrete character history. In fact, this plotted history is just one stochastic realization of many plausible histories, sampled in direct proportion to its Bayesian (or empirical Bayesian) posterior probability. Stochastic character mapping needs to be performed repeatedly (say, 100 or 1,000 times) to obtain a representative sample from the posterior distribution of plausible histories for our character; however, this creates the difficulty of having to somehow visualize in aggregate the results from many maps. If the character has changed state only once or a few times in the history of our tree, then the variability among stochastic reconstructions will generally be relatively small. In this case, any single stochastic map will be quite similar to the average map, and consequently, there may be no need to aggregate our visualization across maps. However, in cases when the character changes state more often, ancestral states at internal nodes and changes along branches will tend to vary much more among stochastically sampled reconstructions. In this case, it could be useful to employ a visualization technique that can incorporate information about this uncertainty.

One way to aggregate the results of many stochastic character histories is to simply compute the posterior probability that each node is in each state represented in our dataset. To do this, we just go through every branch of the tree and find the end state of that branch. The relative frequency of each character state for the ending state of each branch (except the root) is our estimate of the posterior probability that the corresponding daughter node is in that state. For the root node, we just pick one of the two daughter edges for each stochastic map simulation and compute the relative frequency that the starting state for that edge is in each state. It does not matter which one, because (again) it is theoretically impossible that the character changes state exactly at the root node in a continuous-time model.

To map these states on the tree, we can first plot our tree (or our tree with a representative stochastic character map) as normal. Then, having stored the horizontal and vertical positions of all internal nodes in memory, we can overlay the posterior probability that the node is in each state as a pie diagram plotted at each internal node of the tree using the same colors as were used in the mapping. The pie charts overlain on the graph of Fig. 4.3 show an example of this type of visualization.

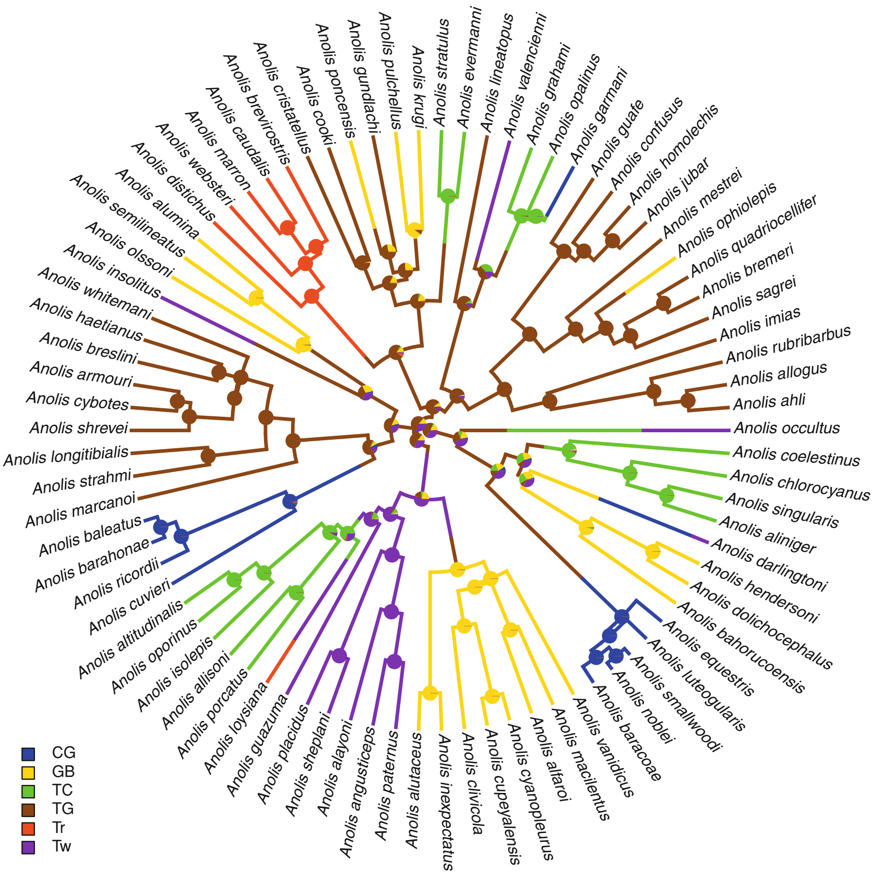

Fig. 4.3 Stochastic character mapping of the multistate discrete character "ecomorph" on the tree of Greater Antillean *Anolis* lizard ecomorph species. Any single map is a stochastic history sampled from the posterior distribution of histories in proportion to its probability. The pie charts at internal nodes show the posterior probabilities aggregated across 100 stochastically mapped character histories using the empirical Bayesian method and a single-rate (i.e., "equal rates") character evolution model. Ecomorphs are named for the microhabitat in which they are most often found, as follows: *CG* crown-giant, *GB* grass-bush, *TC* trunk-crown, *TG* trunk-ground, *Tr* trunk, *Tw* twig

4.3.3 Aggregating Stochastic Maps: Branch Posterior Density Mapping

If our discrete character is a binary (i.e., two-state) character, then we have a further option for plotting that does not completely dispose of the information in our stochastic maps about where along branches our character changes state. Instead, we can map the posterior probabilities of our binary discrete character continuously along the branches and nodes of our phylogeny (Revell 2013). Figure 4.4a gives an illustration of the method using data for the feeding modes of biting vs. suction feeding in elopomorph eels (Collar et al. in revision).

To create this plot, I first finely segmented each branch of the maximum clade credibility tree from a Bayesian phylogeny inference posterior sample of 1,000 phylogenies. Then I went through all the stochastic character maps and asked whether each corresponding segment was in state *0* or state *1*. I tallied the relative frequency of each segment being entirely in state *0* or *1*, and for the instances in which the state changed within a segment, I computed a weighted tally with the weights being set equal to the fraction of time spent in each state for that segment. Having computed the posterior probability as a (near) continuous function of branch position, I then generated a color map and plotted the edges of the tree colored by this map. This method is implemented in phytools (Revell 2012, 2013).

We can use a similar approach to visualize the sampling variance across stochastic character maps in our posterior sample. For a binary character, this variance is equal to $p(1-p)$, where p is the computed posterior probability that the character is in state *1* (vs. state *0*). If the character changes frequently on the tree, this will result in substantial variability in stochastic character histories. By contrast, if the character changes only once or a small number of times in the tree, then all stochastic maps will tend to be similar, and there will be much less variability among stochastic maps in the posterior sample. In Fig. 4.4b, I've created a plot showing the same phylogeny as in Fig. 4.4a with variability among stochastic reconstructions mapped along the branches in grayscale (from white being highly uncertain to black being highly certain). Note that the general pattern is that nodes and edges deep in the tree are uncertain, whereas nodes and edges close to the tips have low variance among maps. Although nodes and edges at the tips of the tree will always tend to have low variance, particularly as we get closer and closer to the tip states (which are known), as a general rule, deeper nodes and edges will be more uncertain if the character state changes frequently—and less so if it changes rarely.

Discerning readers may notice that no additional information is contained in Fig. 4.4b relative to Fig. 4.4a since any branch with a posterior probability of state *1* (biting) that is close to 0.0 or 1.0 will have low uncertainty (black), whereas any branch with intermediate posterior probability of being in state *1* will have high uncertainty (white). This is indeed correct. In fact, Figs. 4.4a and b are just different visualization of the same data, so it would be at the authors' discretion which is more appropriate to their study. It's interesting to consider that the conditions under which we can map the uncertainty of the posterior density from stochastic mapping on a tree are broader than the conditions in which we can map the posterior density itself. Specifically, it's technically challenging to map the posterior density for more than two (or perhaps three, see Revell 2013) stochastically mapped character states on the tree, whereas the difficulty of mapping the uncertainty among maps is not influenced by the number of states for our trait.

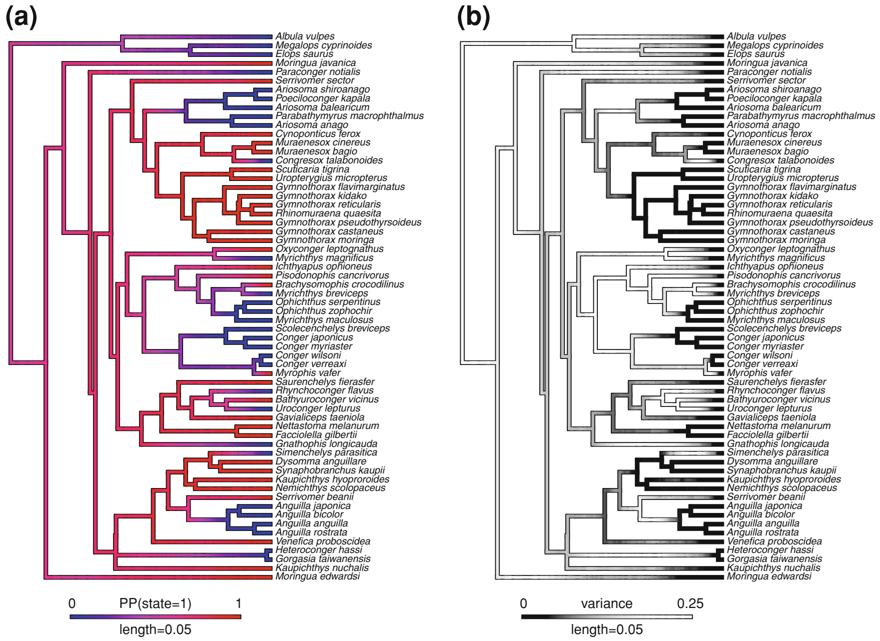

Fig. 4.4 Phylogeny of elopomorph eels from Collar et al. (in revision). **a** Posterior density of feeding mode "biting" versus "suction feeding" mapped on the tree from 100 stochastic character maps. **b** Variance among maps for feeding mode. Black indicates low variance (high certainty in the reconstructed trait value), whereas white indicates high variance

4.4 Continuous Character Methods

4.4.1 The Traitgram

One of the simplest approaches for visualizing continuous trait data on a tree is projection of the tree into a two-dimensional space defined by time and the continuous trait of interest. This has been called a "traitgram" (e.g., Ackerly 2009; Evans et al. 2009), and an example from simulated data is given in Fig. 4.5a.

The procedure to create a traitgram in this style is as follows. First, we estimate the ancestral states for our phenotypic trait at all the nodes of the tree (Schluter et al. 1997). Next, we compute for each node the height of the node above the root of the tree using a pre-order tree traversal. Then, we plot all nodes and tips with a vertical position determined by their known or estimated trait values and a horizontal position determined by the height above the root for that node or tip. Finally, we connect all parent and daughter nodes by edges. In this type of visualization, it's important to keep in mind only the horizontal dimension of edge length contains information about branching times in the tree.

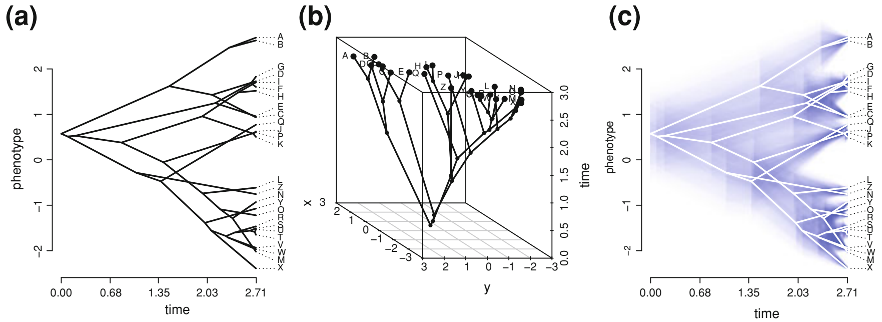

Fig. 4.5 **a** Hypothetical "traitgram" (projection of the tree into a space defined by time and the continuous trait) for a simulated, 26-taxon tree. **b** Simulated three-dimensional traitgram (two phenotypic trait axes plus time). **c** The traitgram from Fig. 4.5a, with 95 % confidence limits around ancestral values shown by increasing transparency in the plotted lines

Usually, we label terminal nodes using the names of the corresponding tip species in the tree. If a number of tips have similar values on the phenotypic trait axis, then this can create a messy plot in which labels overlap and are unreadable. In the example of Fig. 4.5a, I used numerical optimization to space the tip labels using a cost function that penalizes both label overlap and distance from the vertical position of the tip node. (We can specify arbitrary costs for each—depending on whether we find label overlap or label vertical displacement more undesirable.)

It is also possible create a three-dimensional traitgram. In this case, the vertical dimension (for example) might be time, whereas the remaining two dimensions show observed or reconstructed trait values for two continuously valued traits. Figure 4.5b shows a static image of a three-dimensional traitgram plotted with time on the vertical axis. Although this is a fixed plot, phytools can also create a three-dimensional plotting object that can be spun or animated using the R package rgl (Adler and Murdoch 2013).

Finally, the traitgram algorithm can be used to create a visualization capturing uncertainty in ancestral character estimation. In this case, we can use the Hessian matrix or the formulae of Rohlf (2001) to compute the standard error and 95 % confidence interval around ancestral state estimates. Having done this, we can plot 95 % confidence limits around ancestral states (and edges in the traitgram) using a continuous color or transparency gradient, such as that illustrated in Fig. 4.5c.

4.4.2 Projection of a Tree Into Morphospace

Another common visualization method is a complete projection of the tree into a two- or three-dimensional morphospace. This visualization is commonly referred to as a "phylomorphospace" plot (e.g., Rohlf 2002; Sidlauskas 2008).

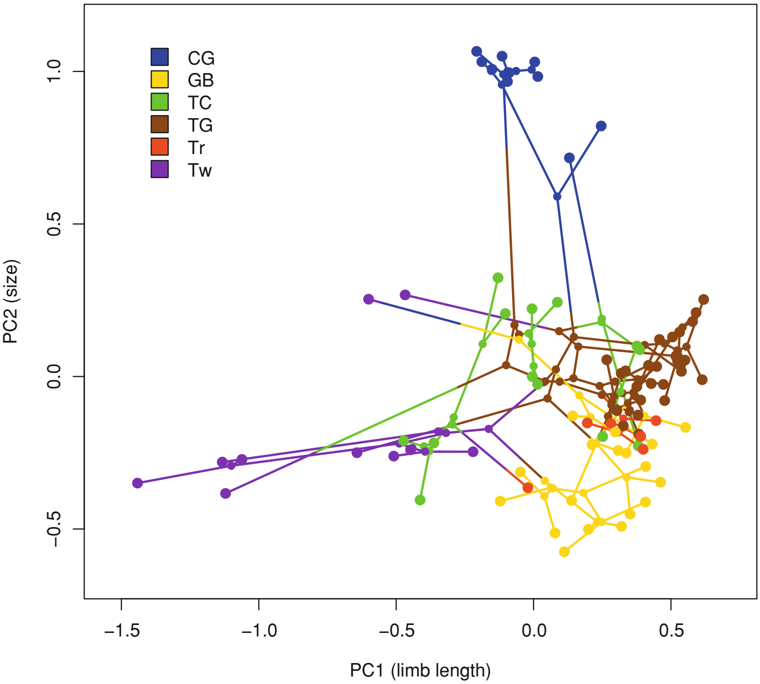

Fig. 4.6 Phylomorphospace (projection of the tree into morphospace) for two principal component axes from 82 species of Greater Antillean *Anolis* lizards. "Ecomorph" class (i.e., ecomorphological habitat specialist) from one stochastic map is projected onto the phylomorphospace. Ecomorphs are as in Fig. 4.3

To create a phylomorphospace visualization, we first need to estimate the ancestral states for all internal nodes in the tree including the root (Schluter et al. 1997). Then, we plot all the estimated states at nodes and the observed states at the tips into our bivariate space. Finally, we connect all parent to daughter nodes with edges and add labels if desired. A visualization of a phylomorphospace plot in two dimensions is given in Fig. 4.6. This figure shows a projection of the phylogeny of greater Antillean ecomorph species of anoles into a two-dimensional principal component morphospace defined by relative limb lengths on the horizontal (PC1) and overall size on the vertical (PC2). Overlain on the projection is a single stochastic map of the evolution of ecomorph on the tree of anoles—the same stochastic map, in fact, as in Fig. 4.3. (See Sect. 4.3.1 for more detail on stochastic character mapping.) The phylogeny and data are from Mahler et al. (2010).

One unfortunate attribute of phylomorphospace visualizations of this type is that all information about time since the root is thrown away during plotting. Recently, some authors (Miller et al. 2013) developed an approach to try and show this information on a phylomorphospace plot using a color gradient that changes continuously from the root to the tips of the tree. Figure 4.7 shows a tree (in panel *a*)

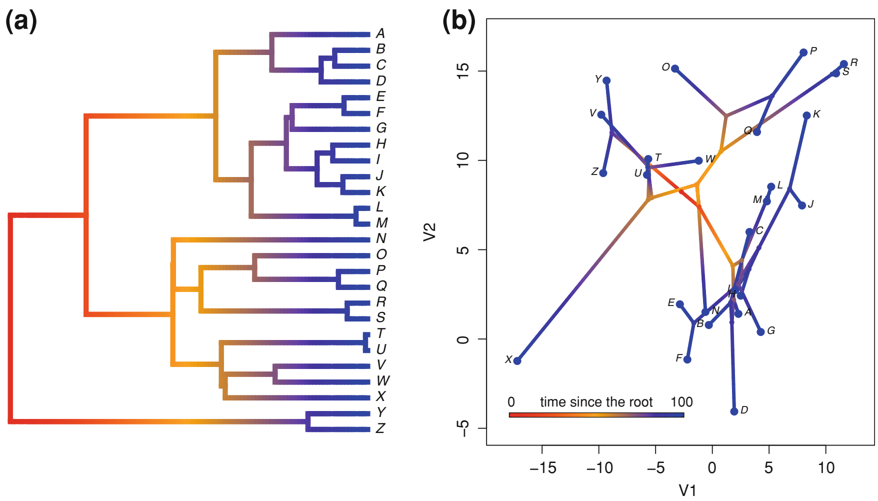

Fig. 4.7 **a** Stochastic phylogeny with time since the root overlain as a color gradient. **b** Simulated phylomorphospace with the color gradient retained (following Miller et al. 2013)

and an example phylomorphospace plotted in this style (Fig. 4.7b), with the temporal dimension retained via a continuous color gradient from the root of the tree (red) toward the tips (blue).

4.4.3 Continuous Character Mapping on the Tree

A final method for continuous character visualization uses the same technique as was described for a posterior density plot from a set of stochastic map trees; however, in this case, we estimate the ancestral states for internal nodes using ML and then interpolate the states along the branches of the tree using Eq. (2) from Felsenstein (1985). Having done this, we are prepared to map our continuous trait on the tree using a continuous color gradient. This method is implemented in phytools (Revell 2012, 2013; also see Verbruggen 2008). An example of this continuous trait mapping is given in Fig. 4.8 using log-transformed body size (snout-to-vent length, or SVL) in Greater Antillean *Anolis* lizards (Mahler et al. 2010). Using the phytools package, it is also possible to combine methods a and b of this section to visualize trait evolution for more than two characters in a single graph. For instance, Fig. 4.9 shows a four-trait multivariate phylogenetic scatterplot matrix for simulated data. The diagonal consists of continuous character maps for each *i*th trait, whereas the *i*, *j*th off-diagonal cell shows a bivariate projection of the tree into morphospace for traits *i* and *j*. For both Figs. 4.8 and 4.9, the specific color scheme is arbitrary, and a different color palette can easily be specified by the user (if, for instance, a more color-blind-sensitive color scheme is desired).

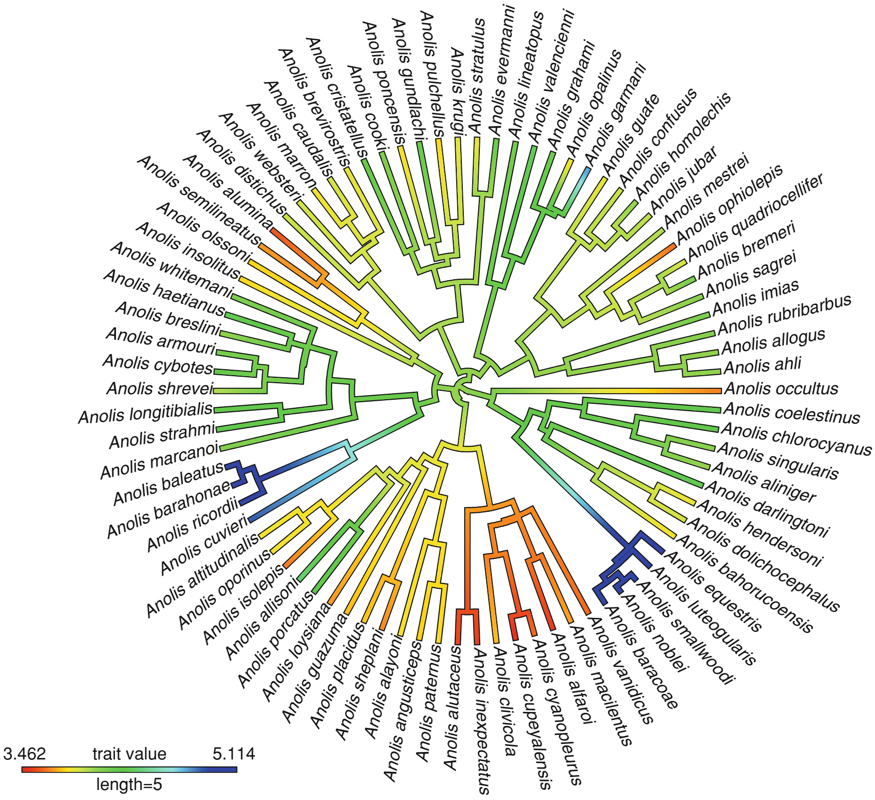

Fig. 4.8 Body size mapped using a continuous color gradient on a phylogeny of 82 Caribbean anole species from Mahler et al. (2010)

4.5 Additional Methods

In Sects. 4.3 and 4.4, I illustrated some different visualization techniques for discrete and continuous character data; however, it is also relatively straightforward to combine some of these techniques into a single plot. I already showed an example of this in Fig. 4.6, which gives a phylomorphospace with an overlain discrete character stochastic mapping.

As these approaches thus far only extend in minor ways the visualizations already shown, I won't dwell extensively on specific methodology; however, by way of illustration, Fig. 4.10a shows a stochastically mapped discrete character overlain on a continuous character traitgram, while Fig. 4.10b shows a bivariate projection of the tree into morphospace with a posterior density from stochastic mapping overlain. This type of plotting method is especially useful in exploratory data analysis for datasets in which (for instance) the state of discrete character is hypothesized to influence the rate of evolution in a continuous character (e.g., in

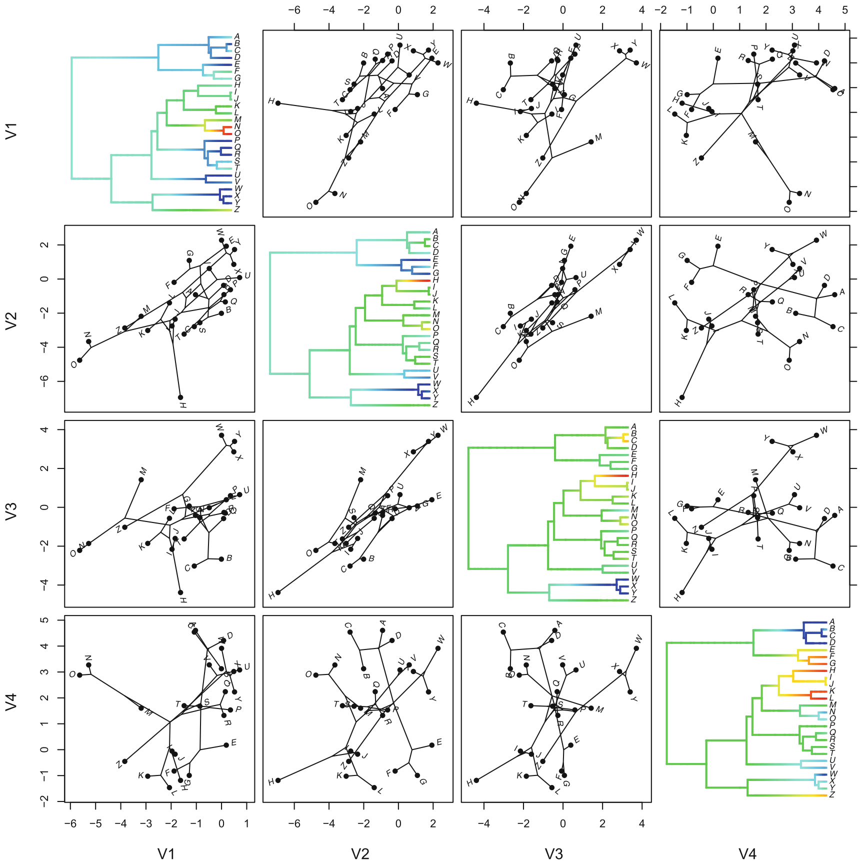

Fig. 4.9 Simulated four-trait phylogenetic scatterplot matrix. Each diagonal element is a continuous character projection on the tree in which red branches indicate small values for the phenotypic trait and blue branches large values, whereas off-diagonals are phylomorphospaces for each *i*, *j*th pair of traits

the simulated data of Fig. 4.10a), or in which the state of a discrete character influences the evolutionary correlation between traits (e.g., Fig. 4.10b).

Finally, trees can also be projected onto a geographic map. For instance, Fig. 4.11a shows a simulated phylogenetic tree in which the tips of the tree point to different geographic localities (perhaps the center of a hypothetical species range or the type locality for the species) on a world map. All the nodes on the tree have been rotated using a “greedy” optimization method to minimize line crossing. The method merely climbs up the tree using a pre-order (root-to-tip) traversal, rotates each node, and accepts the rotation if it reduces the objective function—which is the difference in rank order between the left-to-right order of the tip labels and the west-to-east ordering on the map. Figure 4.12b shows a different type of direct projection of the tree onto a map; however, in this case, it is

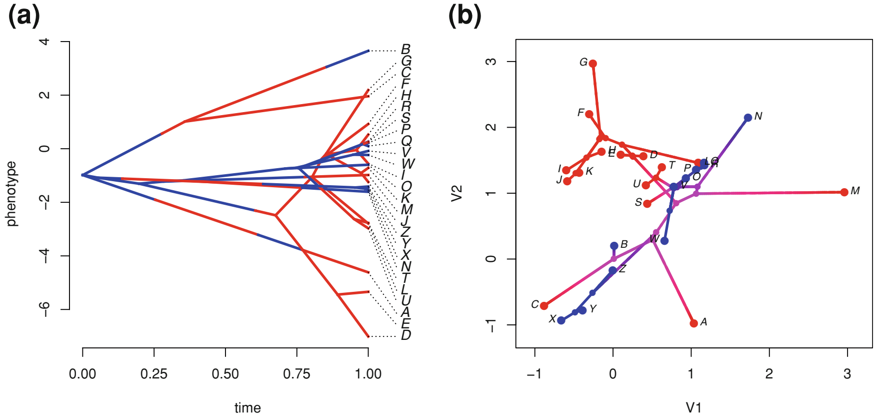

Fig. 4.10 **a** Traitgram on a simulated tree with a stochastic character map overlain. The continuous character data were simulated with a high rate of evolution on the red branches of the tree and a low rate on the blue branches. **b** Phylomorphospace with a posterior density map from 100 stochastic character maps overlain. Data were simulated with a high evolutionary correlation on the blue branches of the tree

important to keep in mind that the locations of the internal nodes in this projection are not equivalent to ancestral range reconstructions (which may be possible to obtain using different methods outside the scope of this chapter, e.g., Ree and Smith 2008).

4.6 Programming Phylogeny Visualization Methods in R

4.6.1 The Structure of a "phylo" Object

The first and most useful thing to understand when developing a plotting method for phylogenies in R is the basic structure of a phylogeny in memory. Phylogenies are stored in R as an object of type *list* with the *class* attribute set to "phylo". Thus, we say that a phylogeny is stored as an *object of class "phylo"*. A list in R consists of a set of objects that can be the same or different in type. For instance, a list could consist of a matrix, a vector of real numbers, and a character string. In this case, the "phylo" object, *tree*, consists of the following four elements and one or more attributes. We can denote the elements of a list using double square brackets (i.e., [[]]) or the dollar sign ($):
tree$edge: *edge* is a matrix of dimensions $e \times 2$ in which e is the number of edges in the tree. For a fully bifurcating tree, $e = 2n - 2$, for n total taxa in the tree. This is just the number of tips (n) plus the number of internal nodes ($n - 1$) minus 1, since every tip or internal node (except for the root) is preceded by an edge. The matrix

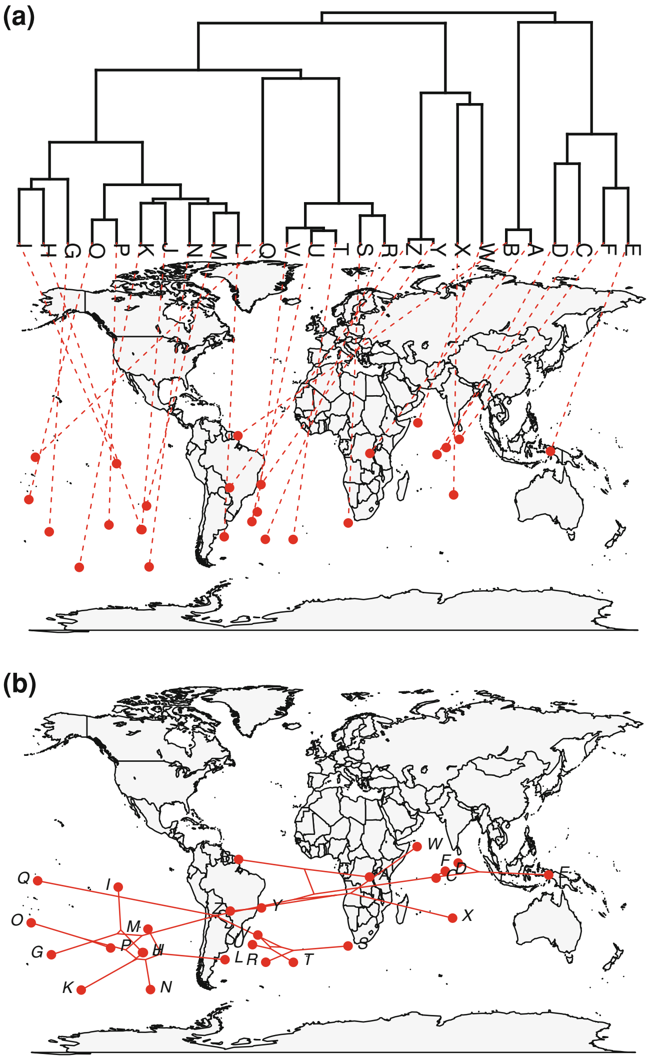

Fig. 4.11 Two different projections of a phylogeny onto a geographic map. **a** A projection in which the tips of the tree are connected to locations on the map via dotted lines. The nodes of the tree were rotated using a "greedy" optimization method to minimize line crossing. **b** A "direct" projection of the phylogeny onto the tree. Note that nodes should not be interpreted as reconstructed ancestral areas in this visualization

tree$edge contains the starting and ending indices (i.e., node number) of every edge in the tree. These indices are given in the boxed numbers of the example five-taxon tree of Fig. 4.12. By convention, indices 1 through n are assigned to the tip nodes in the tree, whereas indices $n + 1$ through $m + n + 1$ (for m internal nodes) are

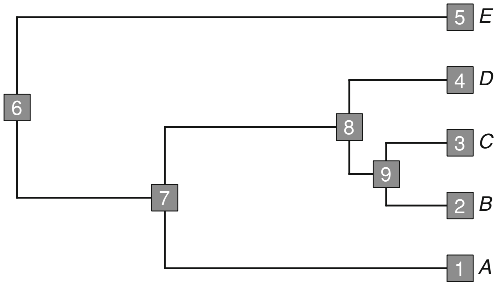

Fig. 4.12 Five-taxon phylogeny with node numbers

assigned to the internal nodes of the tree. For the tree in Fig. 4.12, an example ordering of *tree$edge* is as follows:

```
> tree$edge
     [,1] [,2]
[1,]    6    7
[2,]    7    1
[3,]    7    8
[4,]    8    9
[5,]    9    2
[6,]    9    3
[7,]    8    4
[8,]    6    5
```

tree$Nnode: *Nnode* is an integer giving the total number of internal nodes in the tree, including the root.

```
> tree$Nnode
[1] 4
```

tree$tip.label: *tip.label* is a vector containing all the tip labels for the tips of the tree. The order of *tree$tip.label* is the index order for the nodes. For instance, for the tree in Fig. 4.12, this is merely as follows:

```
> tree$tip.label
[1] "A" "B" "C" "D" "E"
```

tree$edge.length: *edge.length* is a vector containing the lengths of all the edges of the tree in the order of the rows of *tree$edge*. In Fig. 4.12, this vector is as follows:

```
> tree$edge.length
[1]  4  8  5  1  2  2  3 12
```

Finally, the “phylo” object has at least one attribute, its class. This is just a string which tells R how to treat the object in certain custom functions built to deal with objects of this type. In this case, the attribute is simply:

```
> attr(tree,"class")
[1] "phylo"
```

Special types of “phylo” objects can have additional elements or attributes.

4.6.2 Plotting a Simple Phylogram

The next thing that I’ll illustrate is how to use the algorithm of Sect. 4.2 to plot a simple, right-facing phylogram. Whereas in Sect. 4.2 I focused on a general algorithm that applies theoretically to any programming language or development environment, here I’ll give specific R code. Obviously, packages like the R phylogenetics libraries ape and phytools already contain numerous functions for drawing trees; however, a basic understanding of how trees are plotted in R may be useful to investigators interested in developing totally new approaches for visualization.

The code I give below *depends* on the R packages ape and phytools. That means that it uses functions internally that belong to those R function libraries. To start, we should load those packages:

```
library(ape)
library(phytools)
```

The first step is to figure out how many tips we have in the tree and then reorder the tree so that the edges of *tree$edge* are “cladewise”—that is, edges in the same clade are next to each other in the matrix:

```
n <- length(tree$tip.label)
cw <- reorder(tree,"cladewise")
```

Next, we want to compute the vertical position of all the edges in our rightward-facing tree. To do this, we assign our cladewise-ordered tip heights 1 through *n* and then compute the heights for all internal edges via one post-order tree traversal:

```
## create vector
y <- vector(length=n+cw$Nnode)
## assign heights for tips
y[cw$edge[cw$edge[,2]<=n,2]] <- 1:n
pw <- reorder(tree,"pruningwise")
nn <- unique(pw$edge[,1])
## assign heights for internal nodes
for(i in 1:length(nn)){
  yy <- y[pw$edge[which(pw$edge[,1]==nn[i]),2]]
  y[nn[i]] <- mean(range(yy))
}
```

Then, we compute the starting and ending points of each edge on the tree. This can be done for a tree in cladewise order using the phytools function *nodeHeights*:

```
X <- nodeHeights(cw)
```

The matrix ***X*** has dimensions equal to *tree$edge,* and every element of ***X*** corresponds to the height above the root of the corresponding element of *tree$edge*.

Now, we are ready to open a new plotting object. Here, we crudely size the horizontal (*x*) dimension of our plotting area to be 10 % larger than the total length of our tree—to allow space for labels. In "real" tree-plotting functions, we would use a more sophisticated algorithm for this to ensure that enough (but not too much) space was allocated for plotting labels:

```
plot.new()
par(mar=rep(0.1,4))
plot.window(xlim=c(0,1.1*max(X)),ylim=c(0,max(y)+1))
```

Next, we can plot all the horizontal lines in our tree. This is easy because for each edge, the *x*-axis coordinates correspond to a row of ***X***. The single *y*-coordinate can be found by matching the endpoint of the edge (i.e., *cw$edge[i,2]* for the *i*th edge) with the vector *y*:

```
for(i in 1:nrow(X)) lines(X[i,],rep(y[cw$edge[i,2]],2),lwd=2,lend=2)
```

Then, we add the vertical lines that show the relationships between taxa. Only internal nodes have vertical lines, so we just go through the indices used for internal nodes. Each time, we find the element of ***X*** and the coordinates from *y* that correspond with the target edge, and plot the following:

```
for(i in 1:tree$Nnode+n)
  lines(X[which(cw$edge[,1]==i),1],range(y[cw$edge[which(cw$edge[,1]==i),2]]),lwd=2,
  lend=2)
```

Finally, we can plot tip labels. This is easy. The vertical position is the position we assigned at the beginning of the exercise; the horizontal position is the corresponding node height of the terminal node for that tip:

```
for(i in 1:n) text(X[which(cw$edge[,2]==i),2],y[i],tree$tip.label[i],pos=4,offset=0.1)
```

Try it!

4.6.3 Plotting a Simple Projection of the Tree Into Morphospace (Phylomorphospace)

In Sect. 4.4.2, I described a method to project a phylogenetic tree into a two-dimensional morphospace (i.e., a phylomorphospace plot). What follows is a bit more detail on how to program this visualization method in R, which, as the reader will see, is even simpler than plotting a phylogram. This function again uses ape and phytools. Since phytools is dependent on ape, simply loading phytools should be sufficient:

```
library(phytools)
```

First, let's calculate how many tips we have and then estimate ancestral states for all internal nodes. The latter is accomplished using the phytools function *fastAnc*:

```
n <- length(tree$tip.label)
x <- c(x[tree$tip.label],fastAnc(tree,x))
y <- c(y[tree$tip.label],fastAnc(tree,y))
```

Now, let us plot the tips and nodes of our tree. For better ease of visualization, let's plot internal nodes with a slightly smaller symbol than tips:

```
plot(x[1:n],y[1:n],cex=1.25,pch=21,bg="black",xlab="x",ylab="y")
points(x[1:tree$Nnode+n],y[1:tree$Nnode+n],cex=1,pch=21,bg="black")
```

Then, add the lines connecting parent and daughter nodes in morphospace.

```
apply(tree$edge,1,function(edge,x,y) lines(x[edge],y[edge]),x=x,y=y)
```

Finally, let's label all terminal nodes:

```
text(x[1:n],y[1:n],tree$tip.label,pos=2)
```

4.7 Conclusions and Future Directions

Phylogenetic comparative methods have become central to evolutionary biology over the past thirty or so years (Miles and Dunham 1993; Freckleton et al. 2002; Losos 2011; Baum and Smith 2013) and have even begun to infiltrate other biological and non-biological disciplines, such as genomics, biological anthropology, and linguistics (e.g., Thornton and Desalle 2000; Atkinson and Gray 2005; Nunn 2011). Many chapters of this book discuss innovative new approaches for data analysis in comparative biology. However, an important—but sometimes overlooked—first and last step in data analysis is often visualization. *First*, because plotting trees and comparative data can alert us to deviations or errors in our data and perhaps suggest methods of study that might be useful for our data or question. For instance, in a continuous character mapping on the tree, a color gradient along an edge of the tree that suggested that a lineage changed from the highest observed value of the trait to the lowest (or vice versa) might inspire us to cross-check the phenotypic trait value in our dataset, or the position of a potentially “rogue” lineage in the tree. *Last*, because presenting persuasive and informative figures can be an important tool in conveying relevant information about our study system, question, and results.

Potential methods for visualizing phylogenies and comparative data are limited only by the scope of our imaginations (e.g., Rosindell and Harmon 2012). In this article, I have concentrated on relatively simple methods implemented in one way or another in the phytools R package (Revell 2012). Some of these were originally devised by me, but others were devised by others and implemented by me (e.g., Rohlf 2002; Sidlauskas 2008; Evans et al. 2009; Miller et al. 2013). Other methods still were devised in a slightly different form by others and adapted by me for R and the phytools package (e.g., Verbruggen 2008). The list of methods described in this chapter is not comprehensive; however, it does sample from a broad swath of approaches for visualization in phylogenetic comparative biology across discrete and continuous character data types. I have not discussed visualization methods that use the tree but no phenotypic trait data for comparative analysis (for instance, lineage-through-time plots; Pybus and Harvey 2000; Harmon et al. 2003). A review of these methods could be the topic of a separate article or book chapter.

One major limitation of the approaches described in this chapter is that they are constrained to the “paper paradigm” (Rosindell and Harmon 2012). That is, they are designed to be printed on a piece of paper. The printed page (or at least an electronic version thereof) continues to be the primary mode of communication in the sciences. However, this medium imposes severe limits on the size and scope of visualizations of comparative data and phylogenies. Phylogenetic datasets can now contain thousands or perhaps even tens of thousands of tips (e.g., Bininda-Emonds et al. 2007; Smith et al. 2009). Most of the methods of this chapter would be ineffective at conveying useful information about phylogenetic comparative data for phylogenies of this size. Future method development in phylogenetic comparative biology should look to move beyond the paper paradigm for solutions in visualizing large phylogenies and multivariable phenotypic datasets.

Phylogenetic comparative biology has grown over the past thirty or so years to assume a central position in evolutionary study (Miles and Dunham 1993; Losos 2011). Along with it have come new challenges in visualizing comparative data on trees. In this chapter, I have discussed a number of novel or newly implemented visualization methods for comparative data and phylogenies. In the future, new approaches must address the challenge of very large phylogenies (e.g., Rosindell and Harmon 2012) and increasingly multivariate phenotypic trait data of modern phylogenetic studies.

Acknowledgments Thanks to L. Z. Garamszegi for inviting me to contribute this chapter. Thanks are also due to D. Collar, L. Mahler, and R. Mehta for contributing the data and trees that I used for some of these visualizations; to D. Collar, L. Harmon, and L. Mahler for lots of discussion of these methods; to D. Bapst, L. Mahler, E. Miller, and M. Pie for suggesting (in one way or another) methods that appear in this chapter; and to L. Mahler, R. Maia, and an anonymous reviewer for reading and commenting on an earlier version of this chapter. This research was funded in part by a grant from the National Science Foundation (DEB 1350474).

References

Ackerly D (2009) Conservatism and diversification of plant functional traits: evolutionary rates versus phylogenetic signal. PNAS 106:19699–19706

Adler D, Murdoch D (2013) rgl: 3D visualization device system (OpenGL). R package version 0.93.935

Archie J, Day WHE, Felsenstein J, Maddison W, Meacham C, Rohlf FJ, Swofford D (1986) Newick tree format. More information: http://evolution.genetics.washington.edu/phylip/newicktree.html

Atkinson QD, Gray RD (2005) Curious parallels and curious connections: phylogenetic thinking in biology and historical linguistics. Syst Biol 54:513–526

Baum DA, Smith SD (2013) Tree thinking: an introduction to phylogenetic biology. Roberts and Company, Greenwood Village

Beaulieu JM, O'Meara BC, Donoghue MJ (2013) Identifying hidden rate changes in the evolution of a binary morphological character: the evolution of plant habit in the campanulid angiosperms. Syst Biol 62:725–737

Becker RA, Wilks AR, Brownrigg R, Minka TP (2013) Maps: draw geographical maps. R package version 2.3-2

Bininda-Emonds OR, Cardillo M, Jones KE, MacPhee RDE, Beck RMD, Grenyer R, Price SA, Vos RA, Gittleman JL, Purvis A (2007) The delayed rise of present-day mammals. Nature 446:508–512

Bokma F (2008) Detection of "punctuated equilibrium" by Bayesian estimation of speciation and extinction rates, ancestral character states, and rates of anagenetic and cladogenetic evolution on a molecular phylogeny. Evolution 62:2718–2726

Bollback JP (2006) SIMMAP: stochastic character mapping of discrete traits on phylogenies. BMC Bioinf 7:88

Butler MA, King AA (2004) Phylogenetic comparative analysis: a modeling approach for adaptive evolution. Am Nat 164:683–695

Collar DC, Wainwright PC, Alfaro ME, Revell LJ, Mehta RS, Biting disrupts integration to spur skull evolution in eels

Core Team R (2013) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna

Eastman JM, Alfaro ME, Joyce P, Hipp AL, Harmon LJ (2011) A novel comparative method for modeling shifts in the rate of character evolution on trees. Evolution 65:3578–3589

Edwards EJ, Osborne CP, Strömberg CAE, Smith SA, C4 Grasses Consortium (2010) The origins of C4 grasslands: Integrating evolutionary and ecosystem science. Science 328:587–591

Evans MEK, Smith SA, Flynn RS, Donoghue MJ (2009) Climate, niche evolution, and diversification of the "bird-cage" evening primroses (Oenothera, sections Anogra and Kleinia). Am Nat 173:225–240

Felsenstein J (1985) Phylogenies and the comparative method. Am Nat 125:1–15

Felsenstein J (1988) Phylogenies and quantitative characters. Ann Rev Ecol Syst 19:445–471

Felsenstein J (2004) Inferring phylogenies. Sinauer Associates, Sunderland

Felsenstein J (2012) A comparative method for both discrete and continuous characters using the threshold model. Am Nat 179:145–156

FitzJohn RG (2010) Quantitative traits and diversification. Syst Biol 59:619–633

Freckleton RP, Harvey PH, Pagel M (2002) Phylogenetic analysis and comparative data: a test and review of evidence. Am Nat 160:712–726

Glor RE (2010) Phylogenetic insights on adaptive radiation. Ann Rev Ecol Evol Syst 41:251–270

Harmon LJ, Schulte JA II, Larson A, Losos JB (2003) Tempo and mode of evolutionary radiation in iguanian lizards. Science 301:961–964

Harvey PH, Pagel MD (1991) The comparative method in evolutionary biology. Oxford University Press, Oxford

Huelsenbeck JP, Nielsen R, Bollback JP (2003) Stochastic mapping of morphological characters. Syst Biol 52:131–158

Lemon J (2006) Plotrix: A package in the red light district of R. R-News 6:8–12

Ligges U, Mächler M (2003) Scatterplot3d: An R Package for visualizing multivariate data. J Stat Softw 8:1–20

Losos JB (2009) Lizards in an evolutionary tree: ecology and adaptive radiation of anoles. University of California Press, Berkeley

Losos JB (2011) Seeing the forest for the trees: the limitations of phylogenies in comparative biology. Am Nat 177:709–727

Mahler DL, Revell LJ, Glor RE, Losos JB (2010) Ecological opportunity and the rate of morphological evolution in the diversification of Greater Antillean anoles. Evolution 64:2731–2745

Miles DB, Dunham AE (1993) Historical perspectives in ecology and evolutionary biology: the use of phylogenetic comparative analyses. Ann Rev Ecol Syst 24:587–619

Miller ET, Zanne AE, Ricklefs RE (2013) Niche conservatism constrains australian honeyeater assemblages in stressful environments. Ecol Lett 16:1186–1194

Near TJ, Bolnick DI, Wainwright PC (2005) Fossil calibrations and molecular divergence time estimates in centrarchid fishes (Teleostei: Centrarchidae). Evolution 59:1768–1782

Nielsen R (2002) Mapping mutations on phylogenies. Syst Biol 51:729–739

Nunn CL (2011) The comparative approach in evolutionary anthropology and biology. University of Chicago Press, Chicago

O'Meara B (2012) Evolutionary inferences from phylogenies: a review of methods. Ann Rev Ecol Evol Syst 43:267–285

O'Meara BC, Ané C, Sanderson MJ, Wainwright PC (2006) Testing for different rates of continuous trait evolution using likelihood. Evolution 60:922–933

Pagel M (1994) Detecting correlated evolution on phylogenies: a general method for comparative analysis of discrete characters. Proc Roy Soc Ser B 255:37–45

Paradis E (2012) Analysis of phylogenetics and Evolution with R, 2nd edn. Springer, New York

Paradis E, Claude J, Strimmer K (2004) APE: analyses of phylogenetics and evolution in R language. Bioinformatics 20:289–290

Pennell MW, Harmon LJ (2013) An integrative view of phylogenetic comparative methods: connections to population genetics, community ecology and paleobiology. Ann NY Acad Sci 1289:90–105

Pybus OG, Harvey PH (2000) Testing macro-evolutionary models using incomplete molecular phylogenies. Proc Roy Soc B 267:2267–2272

Ree RH, Smith SA (2008) Maximum likelihood inference of geographic range evolution by dispersal, local extinction, and cladogenesis. Syst Biol 57:4–14

Revell LJ (2012) Phytools: an R package for phylogenetic comparative biology (and other things). Methods Ecol Evol 3:217–223

Revell LJ (2013) Two new graphical methods for mapping trait evolution on phylogenies. Methods Ecol Evol 4:754–759

Revell LJ (2014) Ancestral character estimation under the threshold model from quantitative genetics. Evolution 68:743–759

Revell LJ, Mahler DL, Peres-Neto PR, Redelings BD (2012) A new method for identifying exceptional phenotypic diversification. Evolution 66:135–146

Rohlf FJ (2001) Comparative methods for the analysis of continuous variables: geometric interpretations. Evolution 55:2143–2160

Rohlf FJ (2002) Geometric morphometrics and phylogeny. In: MacLeod N, Forey PL (eds) Morphology, shape, and phylogeny. CRC Press: Boca Raton, pp 175–193

Rosindell J, Harmon LJ (2012) OneZoom: a fractal explorer for the tree of life. PLoS Biol 10:e1001406

Schliep KP (2011) Phangorn: phylogenetic analysis in R. Bioinformatics 27:592–593

Schluter D, Price T, Mooers AØ, Ludwig D (1997) Likelihood of ancestor states in adaptive radiation. Evolution 51:1699–1711

Sidlauskas B (2008) Continuous and arrested morphological diversification in sister clades of characiform fishes: a phylomorphospace approach. Evolution 12:3135–3156

Smith SA, Beaulieu JM, Donoghue MJ (2009) Mega-phylogeny approach for comparative biology: an alternative to supertree and supermatrix approaches. BMC Evol Biol 9:37

Thornton JW, DeSalle R (2000) Gene family evolution and homology: genomics meets phylogenetics. Ann Rev Genomics Hum Genet 1:41–73

Verbruggen H (2008) TreeGradients. Available: http://www.phycoweb.net/software/TreeGradients/

Yang Z (2006) Computational molecular evolution. Oxford University Press, Oxford