Chapter 1
Introduction

1.1 Why Phylogenetics and Functional Traits in Ecology?

The distribution of biodiversity is a, if not the, major focus of ecologists. Specifically, ecologists often investigate the spatial or temporal trends in biodiversity levels within a particular study region or across the planet. The study of biodiversity has traditionally focused on quantifying patterns of species diversity or species richness across some type of gradient and determining the potential processes that have produced the observed pattern. This approach is a cornerstone of ecological investigations and thinking regarding biodiversity. However, there are two clear limitations to this species-centric approach. First, biodiversity is not simply species diversity. Biodiversity also includes the phylogenetic, genetic, and functional diversity in an assemblage [1]. Indeed, species diversity may even be the least informative of all of these dimensions of biodiversity. For example, regions could have the same exact species diversity, but very different levels of phylogenetic and functional diversity and therefore very different levels of biodiversity. Or they could have very similar levels of functional and phylogenetic diversity despite large differences in their species richness [2–5]. Thus, attempting to determine the processes that produce biodiversity cannot be obtained by examining only one component of biodiversity. A second challenge for the species-centric approach to studying biodiversity that is perhaps more important than the first one is that species names are relatively information poor. While they are fundamental to biology, they convey little information regarding the function or evolutionary history of species, and such information is critical for determining the processes that have combined to produce the observed levels of biodiversity. These inherent limitations of a species-centric approach suggest that a more pluralistic approach to studying biodiversity is needed in order to obtain a mechanistic understanding of how patterns of biodiversity are formed [6–13]. In particular, a biodiversity synthesis will necessarily require the consideration of the interrelationships between the three primary components of biodiversity—species diversity, functional trait diversity, and phylogenetic diversity [1]. Ecologists are now embracing this reality and have altered their research programs accordingly.

N.G. Swenson, *Functional and Phylogenetic Ecology in R*, Use R!,
DOI 10.1007/978-1-4614-9542-0_1,

The number of phylogenetic- and functional trait-based analyses in ecology has skyrocketed in recent years resulting in hundreds of publications. Indeed, entirely new fields in ecology have formed such as community phylogenetics, and new grant programs have sprung up such as the United States National Science Foundation's Dimensions of Biodiversity program.

Coinciding with the increased interest in quantifying phylogenetic and functional diversity in ecology, a dizzying array of tools and methods has been generated to incorporate phylogenetic and functional information into traditional ecological analyses. Increasingly, these tools are being implemented in R making them easily and freely accessible to researchers around the planet. The goal of this volume is to lead beginning or advanced R users through phylogenetic- and functional trait-based ecological analyses in R. It is expected that beginning users can use this volume as a step-by-step entryway into phylogenetic and functional analyses for ecology in R, whereas it is expected that more advanced users will be able to use this volume as a "cookbook" or quick reference to understand particular analyses. The volume starts with chapters on the R environment and phylogenetic data in R. These are followed by three chapters providing comprehensive coverage of phylogenetic and functional metrics of biodiversity and one chapter on null modeling and randomizations for phylogenetic and functional trait analyses in R. Lastly, two chapters focusing on integrating phylogenetic and functional trait information are provided followed by a final chapter that focuses on interfacing the R environment with a commonly used C-based program called Phylocom that has been influential in phylogenetic ecology [14].

1.2 Why R?

After learning how to ask fundamentally important questions, basic natural history and field identification of the organisms in their study system, I think there are few skills more useful for young ecologists to learn than programming in general and statistical programming in particular. Ecology, like many other disciplines, is rapidly advancing in its analytical complexity and its utilization of "big data." Performing advanced analyses, even on small datasets, or performing even simple analyses on large datasets typically require some level of comfort with computer code. Indeed, when I meet ecology undergraduate and graduate students (and faculty) at other universities I am often asked whether it is "worth it" to learn a programming language like R. I usually provide the response "of course." In many cases I am met with an unconvinced look. I can read in their eyes that they really don't buy my response as a reason to go through what seems to be a daunting process of learning a computer language. To combat this response I often like to first say learning R is very liberating as it frees one up to do many more analyses that they can currently perform. Second, to buoy the first statement I convey an estimate of the percent of the journal articles that I have published that I think would have been possible without R or the ability to program in some other language. The percent I generally

estimate is surprisingly small to many (<20 %). In other words, a lot of the work I do simply would never be possible without even a basic ability to write computer code. I was fortunate enough to be confronted with this reality very early on in my graduate career while working with big datasets and I realized that I better learn a programming language quick if I was to finish my Ph.D. in under a decade.

Learning R and using it day to day (when not in the field, but often also using it in the field letting analyses run on my computer in the field station while I was out staring at trees) was perhaps one of the most valuable tools I gained in graduate school. While it is certainly valuable to learn other programming languages, I would argue learning R is the best starting place for ecologists. This is because ecology and many other disciplines have converged on R for their statistical analyses. This creates a positive feedback loop where more and more researchers perform their analyses in R and write analytical code specifically for R, and therefore more researchers find themselves drawn into the R universe and also contribute. The R code that researchers produce is often made available in packages or in the supplemental material of journal articles making analyses transparent and widely accessible. The issue of accessibility brings us to another important reason why R should be used. R is free! You do not have to pay large sums of money to run your statistics and neither do your collaborators with whom you would like to share your code. Anyone anywhere can freely download the software on their computer and run the most current and advanced analytical code in their field. This greatly levels the analytical playing field for ecology and that can only be a good thing for our science and for achieving our common goals. So I ask you—why not R? The most advanced ecological analyses are now generally coded in R, and this code is becoming or already has become the common analytical currency in ecology.

1.3 Structure and How to Use This Book?

The book is designed to introduce you to phylogenetic and functional trait analyses that can be performed in R. I will not describe in detail each chapter here, but if you are new to R and/or new to phylogenies you should not skip past Chap. 2. This chapter introduces you to phylogenetic data in R—how it is structured and how it can be plotted and manipulated to meet your research goals. This chapter is simply designed as a primer for ecologists and not a comprehensive treatment on phylogenetics in R. At the given time there are enough R packages for phylogenetic analyses that such a treatment would be hard to compile, but I do highly recommend *Analysis of Phylogenetics and Evolution in R* by Emmanuel Paradis in the Springer UseR! series [15] as a wonderful introduction to phylogenies in R and comparative analyses. I would also highly recommend reading a few key texts regarding phylogenetics and comparative methods to help you fully understand what goes into inferring phylogenies and analyzing data in a phylogenetic context (e.g., [16–18]). The present book will cover some similar topics covered in the Paradis book [15] related to how to handle and plot phylogenetic data in Chap. 2 and comparative

analyses in Chap. 7, but the result of the present book is a significant departure focusing primarily on ecological analyses and not macroevolutionary analyses per se. Similarly, the UseR! series book *Numerical Ecology in R* by Daniel Bocard, Francois Gillet, and Pierre Legendre [19] would likely be of interest and use to the readers of this book for general ecological analyses, but the present book significantly differs due to its exclusive focus on phylogenetic and trait data.

The vast majority of the analyses to be discussed in this book can be accomplished using simple "plug and chug" functions in a variety of existing R packages. In teaching courses and workshops on these topics, I have come to two main conclusions. The first is that the participants in my courses and workshops often find it very difficult to navigate the large number of packages available, and they find it difficult to determine whether certain functions do what they want or are similar to other functions in other packages. The second is that participants in my courses and workshops can very easily type in a line of code and get a result, but learning to do this is not very beneficial by itself. This is because the student doesn't realize exactly what was calculated and how it was done. This causes serious problems and limitations when the time comes to interpret the results or when a researcher decides they would like to modify the analytical approach to suit their particular needs. With these issues in mind, the majority of the code provided in the book is designed to lead you, the reader, through the computational steps necessary to calculate the metrics being discussed. I will use my own code to achieve this goal. In some cases my code will be very similar to that in the "canned" R functions already available, and in other cases the code may be significantly different but produces the same result. This difference can be due to different coding styles between me and the original author or my attempts to speed up code by using functions like `apply()` instead of `for()`. After we have broken an analysis into its individual components and calculated the desired result, I will provide you with the name of the "canned" function in an R package, where possible, that should provide the same result. Thus, if you wish to eschew learning how the fine details regarding how the analysis works and what it means, you can ultimately just use the functions highlighted at the end of each section. Though, I don't recommend this approach and I hope that you read, work, and think through the components of the code I provide so you have a detailed understanding of how that number you receive at the end was calculated and what it means. By doing this you will also learn R and learn how to tinker with R code for phylogenetic and functional trait analyses so that you can customize new analyses for your own particular dataset. While working through the examples in the book, you may find that you often run across a new function that you were not aware of before and you may want a more detailed description of what is provided to that function and what comes out of it. To get this information you can access the help file for any function using a "?" and then the function name. For example, if you wanted to know what the `mean()` function does in R, you could see the help page for this function by typing:

```
> ?mean
```

I highly recommend taking this approach and I often do it myself to navigate the code of others and to remind myself how to use a function that I use infrequently.

At the end of each chapter you will find a series of exercises. Some of the exercises will be quite simple and are simply designed to get you used to running the analysis on a variety of datasets. Other exercises will require you recall the information you learned in previous chapters. The goal of this is to help you integrate concepts and information and to help you memorize code and analyses through repetition in different venues. A few exercises will require you to use functions we have not covered. These exercises will be much more difficult to accomplish for the new user, but I have generally told you what new functions you will likely have to use to accomplish the task. It is then up to you to discover how to use these functions and put them together to solve the problem. This is the type of practical problem you will encounter in your future work where you have a particular problem to solve; you break that large problem into many small problems that can be solved with the right tools (i.e., R functions) and then you integrate the solutions to all of those problems to solve the one large problem. Going through advanced exercises such as these will rapidly help you become a more powerful R user.

Lastly, the book relies on many example datasets. Some of these datasets are subsets of larger datasets I utilize for my own work in plant ecology. Others are datasets I have "cooked up" in R. I encourage you to first utilize these datasets to run the analyses, but you should then quickly transition to using your own datasets. As you will quickly find out it is very easy to plug and chug with example datasets, but tiny problems will lurk when you use your own dataset. While these issues regarding minor differences in formatting between files, for example, may be frustrating, it is a common obstacle in data analysis and learning to confront these problems sooner rather than later will be useful to you.

1.4 Setting Working Directories and Package Installation

This book is intended for an audience that spans researchers that are relatively new to R to more advanced R users all of whom would like to incorporate phylogenetic and/or functional information into their research programs. In order to span this gradient, it is necessary to cover some basics that an advanced user does not need to review. For those advanced users this subsection will not be that useful aside from the list of packages at the end that we will utilize in this book. Relatively, new R users may need this section for a brief review on what working directories are and how they are set and how R packages are installed and loaded. We will begin with discussing working directories.

The working directory is the folder (a.k.a. directory) on your hard drive in which the files that you are using and creating are stored. For example, you have a file for your phylogenetic tree and a file for your community data in a folder and you would like to read those files into R and generate output in R that you can write to this same folder.

This can be accomplished by typing in the path to your files every time you read and write them into and out of R, but it is often easier to simply set a single working directory for the project you are working on at that time. For those of you still not totally excited about typing in commands, you can set the working directory in R using drop down menus:

For PC it is under the "File" menu as "Change dir…"
For MAC it is under the "MISC" menu as "Change Working Directory…"

For those of you ready to take the plunge and start typing commands, you can set working directories if you know the "path" to your file as. For example:

For PC:

```
> setwd("C:/my.working.directory/")
```

For Mac:

```
> setwd("/Users/Nate/my.R.project.folder")
```

You can always find out your current working directory by typing:

```
> getwd()
```

Getting the current working directory can be a good way to find out "where you are" currently so you can set a new path for your desired working directory. If you are using the drop down menu to set your working directory, it is useful to get the current working directory path using `getwd()` so you can begin to learn what a path looks like and how to define one. Once you have set your working directory you can obtain a list of all of the files in that working directory. You can accomplish this for any directory, actually, on your computer, but for your current working directory you can simply type:

```
> list.files(getwd())
```

You will now see a list of file names print out on your R console. These are all of the files currently contained in your working directory.

We are almost ready to jump right in and proceed to the next chapter on phylogenetic data in R, but first we must discuss packages. R packages contain a series of functions that can be used for data manipulation or analysis. The functions in a package may use each other and often other functions written in other packages. That is, a function X in package A may need to perform a sub-analysis that can be performed by function Y in package B. In those instances function X in package A "depends" on function Y in package B. Such dependencies are commonplace and one of the wonderful things about R such that you don't have to write your own function anew for your package. You can simply call a function from another package. In this book we will use many packages for phylogenetic and trait analyses in ecology that are useful by themselves or integrated with other packages.

To install a single package, in this example the R package *vegan*, you can type the following command:

```
> install.packages("vegan", dependencies = TRUE)
```

After typing in this command and hitting return you may be asked to select a "mirror." Simply select a mirror that is closest to your geographic location. This will then be the default location from where you download your desired R packages for the current session. Once you have downloaded an R package, it is there for each R session in the future, but its functions are not available to you immediately until you load the package into memory. To load an R package at the start of your session or midway through a session when you need a new function from a different package, you can use the following code again using *vegan* as our example:

```
> library(vegan)
```

If the package you are loading depends on other packages that are already downloaded on your computer, the other packages will also be loaded. If those packages necessary are not on your computer you will receive a message telling you that they must be downloaded.

Now that we have covered how to install and load specific packages, I will simply list the packages that we will use in this book. These could be all installed en masse, but since you may not want to utilize all analyses in this book or you may not want these many items downloaded to your hard drive I will only list them and expect that you can download those that you need or want when you see me call the library in the code in a chapter. The specific packages we will use are: *ape*, *vegan*, *phytools*, *geiger*, *abind*, *picante*, *Rsundials*, *nlme*, *adephylo*, *phylobase*, *ecodist*, *ade4*, *bipartite*, *geometry*, *packfor*, *GUniFrac*, *SDMtools*, *fBasics*, and *FD*.