Journal of FISH BIOLOGY
Published for the Fisheries Society of the British Isles

Journal of Fish Biology (2012) **81**, 2030–2039
doi:10.1111/j.1095-8649.2012.03464.x, available online at wileyonlinelibrary.com

rfishbase: exploring, manipulating and visualizing FishBase data from R

C. Boettiger*†, D. T. Lang‡ and P. C. Wainwright*

*Center for Population Biology, University of California, Davis, CA 95616, U.S.A. and ‡Department of Statistics, University of California, Davis, CA 95616, U.S.A.

(Received 23 April 2012, Accepted 22 August 2012)

This article introduces a package that provides interactive and programmatic access to the FishBase repository. This package allows interaction with data on over 30 000 fish species in the rich statistical computing environment, R. This direct, scriptable interface to FishBase data enables better discovery and integration essential for large-scale comparative analyses. This article provides several examples to illustrate how the package works, and how it can be integrated into phylogenetics packages such as ape and geiger.

Key words: data access; programmatic; tutorial; XML.

INTRODUCTION

FishBase (www.fishbase.org) is an award-winning online database of information about the morphology, trophic ecology, physiology, ecotoxicology, reproduction and economic relevance of the world's fishes, organized by species (Froese & Pauly, 2012). This repository of information has proven to be a profoundly valuable community resource and the data have the potential to be used in a wide range of studies. Assembling sub-sets of data housed in FishBase for use in focused analyses can, however, be tedious and time-consuming. To facilitate the extraction, visualization and integration of these data, the rfishbase package has been written for the R language for statistical computing and graphics (R Development Core Team; www.r-project.org). R is a freely available open source computing environment that is used extensively in ecological research, with a large collection of packages built explicitly for this purpose (Kneib, 2007).

The rfishbase package is dynamically updated from the FishBase database, describes its functions for extracting, manipulating and visualizing data, and then illustrates how these functions can be combined for more complicated analyses. Lastly, it illustrates how having access to FishBase data through R allows a user to interface with other resources such as comparative phylogenetics software. The purpose of this article is to introduce rfishbase and illustrate core features of its functionality.

†Author to whom correspondence should be addressed. Tel.: +1 610 389 6087; email: cboettig@ucdavis.edu

ACCESSING FISHBASE DATA FROM R

In addition to its web-based interface, FishBase provides machine-readable XML files for 30 622 (as accessed on 14 May 2012) of its species entries to facilitate programmatic access of the data housed in this resource. As complete downloads of the FishBase database are not available, the FishBase team encouraged the use of these XML files as an entry point for programmatic access. Here, caching and pausing features are introduced into the package to prevent access from over-taxing the FishBase servers. While FishBase encourages the use of these data by programmatic access (R. Froese, pers. comm.), users of rfishbase should respect these load-limiting functions, and provide appropriate acknowledgement. A more detailed discussion of incentives, ethics and legal requirements in sharing and accessing such repositories can be found in the respective literature (Costello, 2009; Fisher & Fortmann, 2010).

The rfishbase package works by creating a cached copy of all data on FishBase currently available in XML format on the FishBase web pages. This process relies on the RCurl (Lang, 2012*a*) and XML (Lang, 2012*b*) packages to access these pages and parse the resulting XML into a local cache. Caching increases the speed of queries and greatly reduces demands on the FishBase server, which in its present form is not built to support direct access to application programming interfaces. A cached copy is included in the package and can be loaded in to R using the command:

```
data (fishbase)
```

This loads a copy of all available data from FishBase into the R list, fish.data, which can be passed to the various functions of rfishbase for extraction, manipulation and visualization. The online repository is frequently updated as new information is uploaded. To get the most recent copy of FishBase, update the cache instead. The update may take up to 24 h. This copy is stored in the specified directory (note that '.' can be used to indicate the current working directory) with the current date. The most recent copy of the data in the specified path can be loaded with the loadCache() function. If no cached set is found, rfishbase will load the copy originally included in the package:

```
updateCache(".")
loadCache(".")
```

Loading the database creates an object called fish.data, with one entry per fish species for which data were successfully found, for a total of 30 622 species.

Not all the data available in FishBase are included in these machine-readable XML files. Consequently, rfishbase returns taxonomic information, trophic description, habitat, distribution, size, life-cycle, morphology and diagnostic information. The information returned in each category is provided as plain text, consequently rfishbase must use regular expression matching to identify the occurrence of particular words or patterns in this text corresponding to the data of interest (Friedl, 2006). Any regular expression can be used in search queries. While these expressions allow for very precise pattern matching, applying this approach to plain text runs some risk of error which should not be ignored. Visual inspection of matches and careful construction of these expressions can help mitigate this risk. Example functions are provided for reliably matching several quantitative traits from these text-based descriptions, which can be used as a basis for writing functions to identify other terms of interest.

Quantitative traits such as standard length (L_S), maximum known age, spine and ray counts and depth information are provided consistently for most species, allowing rfishbase to extract these data directly. Other queries require pattern matching. While simple text searches within a given field are usually reliable, the rfishbase search functions will take any regular expression query, which permits logical matching, identification of number strings and much more. The interested user should consult a reference on regular expressions after studying the simple examples provided here to learn more.

TOOLS FOR DATA EXTRACTION, ANALYSIS AND VISUALIZATION

The basic tool for data extraction in rfishbase is the which_fish() function. This function takes a list of FishBase data (usually the entire database, fish.data or a sub-set thereof, as illustrated later) and returns an array of those species matching the query. This array is given as a list of true or false values for every species in the query. This return structure has several advantages which are illustrated below.

Here is a query for reef-associated fishes (mention of 'reef' in the habitat description), and second query for fishes that have 'nocturnal' in their trophic description:

```
reef <- which_fish("reef", "habitat", fish.data)
nocturnal <- which_fish("nocturnal", "trophic", fish.data)
```

One way these returned values are commonly used is to obtain a sub-set of the database that meets these criteria, which can then be passed on to other functions. For instance, if the scientific names of these reef fishes are needed, the fish_names function can be used. Like the which_fish function, it takes the list of FishBase data, fish.data as input. In this example, just the sub-sets that are reef affiliated are passed to the function:

```
reef_species <- fish_names(fish.data[reef])
```

Because the present reef object is a list of logical values (true or false), this can be combined in intuitive ways with other queries. For instance, names of all fishes that are both nocturnal and not reef associated can be queried:

```
nocturnal_nonreef_orders <- fish_names(fish.data[nocturnal & !reef], "Class")
```

Note that in this example, it is also specified that the user wants the taxonomic Class of the fishes matching the query, rather than the species names. fish_names will allow the user to specify any taxonomic level for it to return. Quantitative trait queries work in a similar manner to fish_names, taking the FishBase data and returning the requested information. For instance, the function getSize returns the length (default), mass or age of the fish in the query:

```
age <- getSize(fish.data, "age")
```

rfishbase can also extract a table of quantitative traits from the morphology field, describing the number of vertebrae, dorsal and anal fin spines and rays,

```
morphology_numbers $<-$ getQuantTraits(fish.data)
```

and extract the depth range (extremes and usual range) from the habitat field:

```
depths <- getDepth(fish.data)
```

A list of all the functions provided by rfishbase can be found in Table I. The rfishbase manual provided with the package provides more detail about each of these functions, together with examples for their use.

The real power of programmatic access is the ease with which it is possible to combine, visualize and statistically test a custom compilation of these data. To do

TABLE I. A list of functions and data objects provided by rfishbase

Function name	Description
familySearch	A function to find all fishes that are members of a scientific family
findSpecies	Returns the matching indices in the data given a list of species names
fish.data	A cached copy of extracted FishBase data
fish_names	Return the scientific names, families, classes or orders of the input data
getDepth	Returns available depth range data
getQuantTraits	Returns all quantitative trait values found in the morphology data
getRefs	Returns the FishBase reference identification numbers matching a query
getSize	Returns available size data of specified type (length, mass, or age)
habitatSearch	A function to search for the occurrences of any keyword in habitat description
labridtree	An example phylogeny of labrids
loadCache	Load an updated cache
updateCache	Update the cached copy of FishBase data
which_fish	Which fish is the generic search function for fishbase a variety of description types

so, it is useful to organize a collection of queries into a data frame. The next set of commands combines the queries made above and a few additional queries into a data frame in which each row represents a species and each column represents a variable:

```
marine <- which_fish("marine", "habitat",~fish.data)
africa <- which_fish("Africa:", "distribution",~fish.data)
length <- getSize(fish.data, "length")
order <- fish_names(fish.data, "Order")
dat <- data.frame(reef, nocturnal, age, marine, africa, length,~order)
```

This data frame contains categorical data (*e.g.* is the fish a carnivore?) and continuous data (*e.g.* mass or age of fish). Data visualization tools in R can be taken advantage of to begin exploring these data. These examples are simply meant to illustrate the kinds of analysis possible and how they would be constructed.

For instance, it is possible to identify which orders contain the greatest number of species, and for each of them, plot the fraction in which the species are marine (Fig. 1).

```
biggest <- names(head(sort(table(order),decr=T),~8))
primary_orders <- subset(dat, order %in%~biggest)
ggplot(primary_orders, aes(order, fill=marine)) + geom_bar() +
# a few commands to customize appearance
geom_bar(colour="black",show_guide=FALSE) +
opts(axis.text.x=theme_text(angle=90, hjust=1, size=6)) +
opts(legend.title=theme_blank(), legend.justification=c(1,0),
        legend.position=c(.9,.6)) +
scale_fill_grey(labels=c("Marine", "Non-marine")) +
xlab(" ") + ylab("Number of species")
```

FishBase data excels for comparative studies across many species, but searching through >30 000 species to extract data makes broad comparative analyses quite time-consuming. Having access to the data in R, such questions can be answered

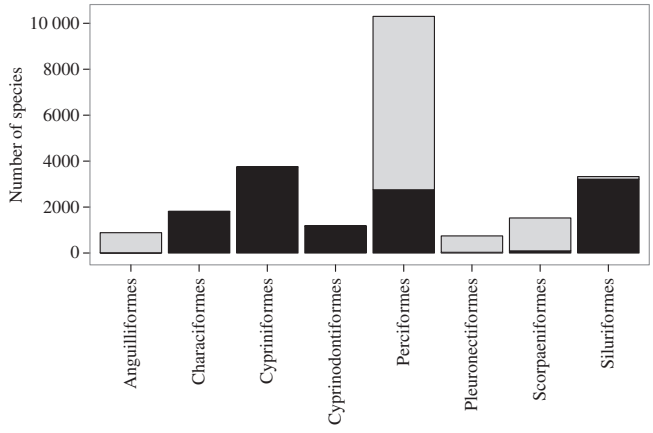

FIG. 1. Fraction of marine species (■, marine; ■, non-marine) in the eight largest orders of teleosts.

as fast as they are posed. Consider looking for a correlation between the maximum age and the size of fishes. Partitioning the data by any variable of interest as well, this example colour codes the points based on whether or not the species is marine associated. The ggplot2 package (Wickham, 2009) provides a particularly powerful and flexible language for visual exploration of such patterns (Fig. 2).

```
ggplot(dat,aes(age, length, shape=marine)) +
  geom_point(position='jitter', size=1) + scale_y_log10() +
  scale_x_log10(breaks=c(50,100,200)) +
  scale_shape_manual(values=c(1,19), labels=c("Marine", "Non-marine")) +
  ylab("Standard length (cm)") + xlab("Maximum observed age (years)") +
  opts(legend.title=theme_blank(), legend.justification=c(1,0),
```

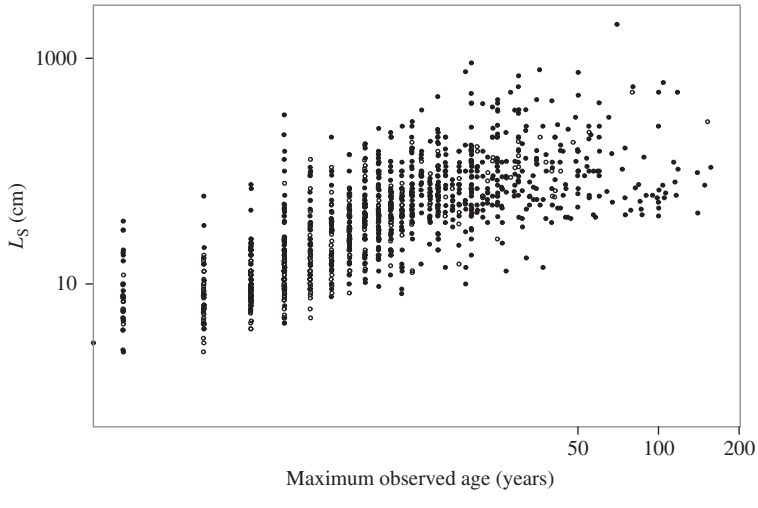

FIG. 2. Scatterplot of maximum age with standard length (L_S) observed in each species (○, marine species; ●, freshwater species).

```
    legend.position=c(.9,0)) +
  opts(legend.key = theme_blank())
```

A wide array of visual displays is available for different kinds of data. A box-plot (Fig. 3) is a natural way to compare the distributions of categorical variables, such as asking whether reef species are longer lived than non-reef species in the marine environment:

```
ggplot(subset(dat, marine)) + geom_boxplot(aes(reef, age)) +
  scale_y_log10() + xlab(" ") + ylab("Maximum observed age (years)") +
  opts(axis.text.x = theme_text(size =~8))
```

In addition to powerful visualizations, R provides an unparalleled array of statistical analysis methods. Executing the linear model testing the correlation of length with maximum size takes a single line:

```
library(MASS)
corr.model <- summary(rlm(data=dat, length ~ age))
```

which shows a significant correlation between maximum age and L_S ($P < 0{\cdot}001$) under a robust linear regression.

COMPARATIVE STUDIES

Many ecological and evolutionary studies rely on comparisons between taxa to pursue questions that cannot be approached experimentally. For instance, recent studies have attempted to identify whether reef-associated clades experience greater species diversification rates than non-reef-associated groups (Alfaro *et al*., 2009). It is possible to identify and compare the numbers of reef-associated species in different families using the rfishbase functions presented above.

In this example, consider the simpler question as to whether there are more reef-associated species in *Labridae* than in *Gobiidae*. Recent research has shown that the families Scaridae and Odacidae are nested within Labridae (Westneat & Alfaro,

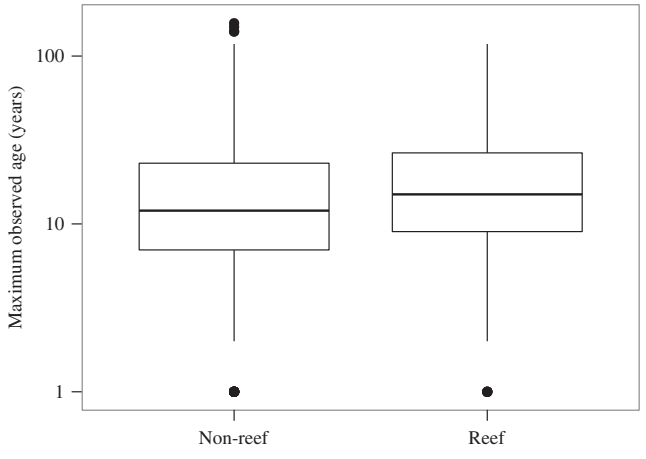

FIG. 3. Distribution of maximum age for reef-associated and non-reef-associated fishes (middle band, median; boxes, 25th and 75th percentiles; vertical lines, ×1·5 the interquartile range; ●, outliers).

2005), although the three groups are listed as separate families in FishBase. All the species in FishBase are obtained from *Labridae* (wrasses), *Scaridae* (parrotfishes) and *Odacidae* (weed-whitings):

```
labrid <- which_fish("(Labridae|Scaridae|Odacidae)", "Family",~fish.data)
```

and get all the species of gobies:

```
goby <- which_fish("Gobiidae", "Family",~fish.data)
```

Identify how many labrids are found on reefs:

```
labrid.reef <- which_fish("reef", "habitat", fish.data[labrid])
labrids.on.reefs <- table(labrid.reef)
```

and how many gobies are found on reefs:

```
gobies.on.reefs <- table(which_fish("reef", "habitat", fish.data[goby]) )
```

Note that summing the list of true or false values returned gives the total number of matches. This reveals that there are 505 labrid species associated with reefs, and 401 goby species associated with reefs. This example illustrates the power of accessing the FishBase data, Gobies are routinely listed as the largest group of reef fishes (Bellwood & Wainwright, 2002), but this is because there are more species in *Gobiidae* than any other family of reef fishes. When the species in each group that live on reefs are counted it is found that labrids are actually the most species-rich family on reefs.

INTEGRATION OF ANALYSES

One of the greatest advantages of accessing FishBase directly through R is the ability to take advantage of other specialized analyses available through R packages. Users familiar with these packages can more easily take advantage of the data available on FishBase. This is illustrated with an example that combines phylogenetic methods available in R with quantitative trait data available from rfishbase.

This series of commands illustrates testing for a phylogenetically corrected correlation between the observed length of a species and the maximum observed depth at which it is found. It begins by reading in the data for a phylogenetic tree of labrids (provided in the package), and the phylogenetics packages ape (Paradis *et al*., 2004) and geiger (Harmon *et al*., 2009):

```
data(labridtree)
library(ape)
library(geiger)
```

Find the species represented on this tree in FishBase:

```
myfish <- findSpecies(labridtree$tip.label,~fish.data)
```

Get the maximum depth of each species and size of each species:

```
depth <- getDepth(fish.data[myfish])[,"deep"]
size <- getSize(fish.data[myfish], "length")
```

Drop missing data, and then drop tips from the phylogeny for which data were not available:

```
data <- na.omit(data.frame(size,depths))pruned <- treedata(labridtree,~data)
```

Use phylogenetically independent contrasts (Felsenstein, 1985) to determine if depth correlates with size after correcting for phylogeny:

```
corr.size <- pic(pruned$data[["size"]],pruned$phy)
corr.depth <- pic(pruned$data[["depths"]],pruned$phy)
corr.summary <- summary(lm(corr.depth ~ corr.size - 1))
```

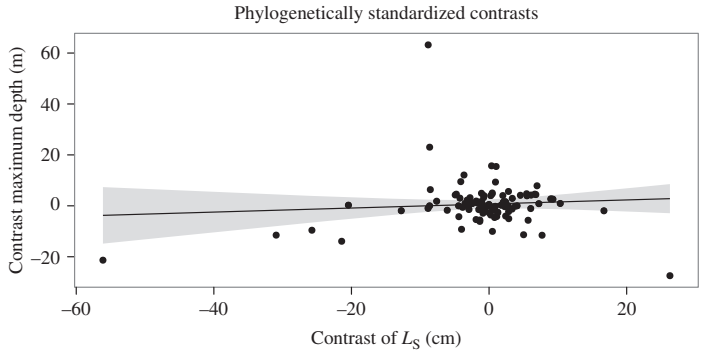

FIG. 4. Correcting for phylogeny, size (standard length, L_S) is not correlated with maximum depth observed in labrids.

which returns a non-significant correlation ($P = 0{\cdot}47$; Fig. 4).

```
ggplot(data.frame(corr.size,corr.depth), aes(corr.size,corr.depth)) +
geom_point() + stat_smooth(method=lm, col=1) +
   xlab("Contrast of standard length (cm)") +
   ylab("Contrast maximum depth (m)") +
   opts(title="Phylogenetically standardized contrasts")
```

Also different evolutionary models for these traits can be estimated to decide which best describes the data:

```
bm <- fitContinuous(pruned$phy, pruned$data[["depths"]], model="BM")[[1]]
ou <- fitContinuous(pruned$phy, pruned$data[["depths"]], model="OU")[[1]]
```

where the Brownian motion model has an AIC score of 1185 while the Ornstein–Uhlenbeck (OU) model has a score of 918·2, suggesting that OU is the better model.

DISCUSSION

With more and more data readily available, informatics is becoming increasingly important in ecology and evolution research (Jones *et al*., 2006), bringing new opportunities for research (Parr *et al*., 2011; Michener & Jones, 2012) while also raising new challenges (Reichman *et al*., 2011). It is in this spirit that the rfishbase package provides programmatic access to the data available on the already widely recognized database, FishBase. Such tools allow researchers to take greater advantage of the data available, facilitating deeper and richer analyses than would be feasible under only manual access to the data. The examples in this article are intended to illustrate how this package works and to help inspire readers to consider and explore questions that would otherwise be too time-consuming or challenging to pursue. This article has introduced the functions of the rfishbase package and described how they can be used to improve the extraction, visualization and integration of FishBase data in ecological and evolutionary research.

THE SELF-UPDATING STUDY

Because analyses using this data are written in R scripts, it becomes easy to update the results as more data becomes available on FishBase. Programmatic access to

data coupled with scriptable analyses can help ensure that research is more easily reproduced and also facilitate extending the work in future studies (Merali, 2010; Peng, 2011). This document is an example of this, using a dynamic documentation interpreter programme which runs the code displayed to produce the results shown, decreasing the possibility for faulty code (Xie, 2012). As FishBase is updated, these results can be regenerated with fewer missing data. Readers can find the original document which combines the source-code and text on the project's Github page (https://github.com/ropensci/rfishbase/tree/master/inst/doc/rfishbase/).

LIMITATIONS AND FUTURE DIRECTIONS

FishBase contains much data that have not been made accessible in machine-readable XML format. Because most of the data provided in the XML comes as plain text rather than being identified with machine-readable tags, reliability of the results is limited by text matching. Improved text-matching queries could provide more reliable information, and facilitate other specialized queries such as extracting geographic distribution details as categorical variables or latitude and longitude co-ordinates. FishBase taxonomy is inconsistent with taxonomy provided elsewhere, and additional package functions could help resolve these differences in assignments.

rfishbase has been available to R users through the Comprehensive R Archive Network (http://cran.r-project.org/web/packages/rfishbase/) since October 2011 and has a growing user base. The project remains in active development to evolve with the needs of its users. Users can view the most recent changes and file issues with the package on its development website on Github, and developers can submit changes to the code or adapt it to their own software.

Programmers of other R software packages can make use of the rfishbase package to make these data available to their functions, further increasing the use and influence of FishBase. For instance, the project OpenFisheries (http://openfisheries.org/) makes use of the rfishbase package to provide information about commercially relevant species.

This work was supported by a Computational Sciences Graduate Fellowship from the U.S. Department of Energy under grant number DE-FG02-97ER25308 to C. B. and National Science Foundation grant DEB-1061981 to P. C. W. The rfishbase package is part of the rOpenSci project (ropensci.org).

References

Alfaro, M. E., Santini, F., Brock, C. D., Alamillo, H., Dornburg, A., Rabosky, D. L., Carnevale, G. & Harmon, L. J. (2009). Nine exceptional radiations plus high turnover explain species diversity in jawed vertebrates. *Proceedings of the National Academy of Sciences of the United States of America* **106,** 13410–13414. doi: 10.1073/pnas.0811087106

Bellwood, D. R. & Wainwright, P. C. (2002). The history and biogeography of fishes on coral reefs. In *Coral Reef Fishes. Dynamics and Diversity in a Complex Ecosystem* (Sale, P. F., ed.), pp. 5–32. San Diego, CA: Academic Press.

Costello, M. J. (2009). Motivating online publication of data. *BioScience* **59,** 418–427. doi: 10.1525/bio.2009.59.5.9

Felsenstein, J. (1985). Phylogenies and the comparative method. *American Naturalist* **125,** 1–15. doi: 10.1086/284325

Fisher, J. B. & Fortmann, L. (2010). Governing the data commons: policy, practice, and the advancement of science. *Information and Management* **47,** 237–245. doi: 10.1016/j.im.2010.04.001

Friedl, J. E. F. (2006). *Mastering Regular Expressions.* New York, NY: O'Reilly Media, Inc.

Jones, M. B., Schildhauer, M. P., Reichman, O. J. & Bowers, S. (2006). The new bioinformatics: integrating ecological data from the gene to the biosphere. *Annual Review of Ecology, Evolution, and Systematics* **37,** 519–544. doi: 10.1146/annurev.ecolsys.37.091305.110031

Kneib, T. (2007). Introduction to the special volume on 'ecology and ecological modelling in R'. *Journal of Statistical Software* **22,** 1–7.

Merali, Z. (2010). Why scientific programming does not compute. *Nature* **467,** 775–777.

Michener, W. K. & Jones, M. B. (2012). Ecoinformatics: supporting ecology as a data-intensive science. *Trends in Ecology and Evolution* **27,** 85–93. doi: 10.1016/j.tree.2011.11.016

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

Parr, C. S., Guralnick, R., Cellinese, N. & Page, R. D. M. (2011). Evolutionary informatics: unifying knowledge about the diversity of life. *Trends in Ecology and Evolution* **27,** 94–103. doi: 10.1016/j.tree.2011.11.001

Peng, R. D. (2011). Reproducible research in computational science. *Science* **334,** 1226–1227. doi: 10.1126/science.1213847

Reichman, O. J., Jones, M. B. & Schildhauer, M. P. (2011). Challenges and opportunities of open data in ecology. *Science* **331,** 703–705. doi: 10.1126/science.1197962

Westneat, M. W. & Alfaro, M. E. (2005). Phylogenetic relationships and evolutionary history of the reef fish family Labridae. *Molecular Phylogenetics and Evolution* **36,** 370–390. doi: 10.1016/j.ympev.2005.02.001

Wickham, H. (2009). *ggplot2: Elegant Graphics for Data Analysis.* New York, NY: Springer.

Electronic References

Froese, R. & Pauly, D. (2012). *FishBase*. Available at www.fishbase.org.

Harmon, L., Weir, J., Brock, C., Glor, R., Challenger, W. & Hunt, G. (2009). *geiger: Analysis of Evolutionary Diversification.* Available at http://cran.r-project.org/package=geiger

Lang, D. T. (2012*a*). *RCurl: General Network (HTTP/FTP/...) Client Interface for R*. Available at http://cran.r-project.org/package=RCurl

Lang, D. T. (2012*b*). *XML: Tools for Parsing and Generating XML within R and S-Plus*. Available at http://cran.r-project.org/package=XML

Xie, Y. (2012). *knitr: A General-purpose Package for Dynamic Report Generation in R*. Available at http://yihui.name/knitr