Bioinformatics, 35(3), 2019, 526–528
doi: 10.1093/bioinformatics/bty633
Advance Access Publication Date: 17 July 2018
Applications Note

Phylogenetics

ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R

Emmanuel Paradis[1,]* and Klaus Schliep[2]

[1]ISEM, IRD, Univ. Montpellier, CNRS, EPHE, 34095 Montpellier, France and [2]Department of Biology, University of Massachusetts Boston, Boston, MA 02125, USA

*To whom correspondence should be addressed.

Associate Editor: Russell Schwartz

Received on March 24, 2018; revised on July 9, 2018; editorial decision on July 11, 2018; accepted on July 13, 2018

Abstract

Summary: After more than fifteen years of existence, the R package ape has continuously grown its contents, and has been used by a growing community of users. The release of version 5.0 has marked a leap towards a modern software for evolutionary analyses. Efforts have been put to improve efficiency, flexibility, support for 'big data' (R's long vectors), ease of use and quality check before a new release. These changes will hopefully make ape a useful software for the study of biodiversity and evolution in a context of increasing data quantity.

Availability and implementation: ape is distributed through the Comprehensive R Archive Network: http://cran.r-project.org/package=ape. Further information may be found at http://ape-package.ird.fr/.

Contact: emmanuel.paradis@ird.fr

Since its first release in August 2002, the R (R Core Team, 2017) package *ape* (Paradis *et al.*, 2004) has developed steadily to provide a set of tools for researchers in evolutionary biology who seek to integrate analyses of DNA sequences, phenotypes and phylogenetic trees. With the release of its version 5.0, *ape* has marked a leap towards a modern software for evolutionary analyses. In the perspective of this release, efforts were put in improving the existing functionalities, providing easy-to-use functions for some basic tasks (e.g. finding data files), and improving the quality checks with respect to the published packages that depend on *ape*.

There is evidence of a substantial number of users of *ape* coming from two sources. First, the number of downloads of *ape* during a few days after a new release has been observed to be around 22 000 (data from https://cranlogs.r-pkg.org/). Second, *ape* has been cited in more than 600 journals from all fields of life and environmental sciences, and some titles in human sciences (data from Web of Science) suggesting it is widely used in the scientific community.

With its version 5.0, *ape* supports all types of phylogenetic trees and networks. The function `read.tree`, already present in the first release of *ape*, can read trees with nodes of degree two (which was not possible before), and the new function `read.evonet` can read phylogenetic networks from extended Newick format files (Cardona *et al.*, 2008). In addition, the R code to read Newick and NEXUS files (which have served during almost fifteen years) has been replaced by a more efficient C code.

ape has its own data object class to store and manipulate DNA sequences (called 'DNAbin'). In the recent versions, we have implemented R 'long vectors' for the 'DNAbin' class so that sequences longer than 2.1 Gb (gigabases) can be read and analysed. The new limit for a sequence length is ≈4.4 Pb ($= 4.4 \times 10^{15}$ bases) which is more than one thousand larger than the currently available data from GenBank and the Genome Projects combined (estimated to be between 2 and 3 Tb). Furthermore, *ape* is able, in theory, to manipulate $\approx 4.4 \times 10^{15}$ sequences each of 4.4 Pb which amounts to a total of $\approx 2 \times 10^{31}$ bases. Clearly, this quantity is much larger than the quantity of active memory of all machines available in the world combined. In addition to these technical improvements, functions to convert the data classes used in BioConductor (Huber *et al.*, 2015) are now provided. To further integrate phylogenetics with the analysis of high-throughput sequencing data, there is also a new function to read FASTQ files (`read.fastq`) to help screen DNA reads and their qualities.

Reading data from files is a common bottleneck in data analysis which becomes crucial in the context of modern genomic data.

Table 1. Computing times (in seconds, averaged over ten replications) of two input functions of *ape* (*n*: number of sequences in the FASTA file or tips in the Newick file, *l*: sequence length, ne: not evaluated)

		`read.dna`		`read.tree`	
n	*l*	*ape* 3.0	*ape* 5.0	*ape* 4.0	*ape* 5.0
10^4	10^3	13.27	0.05	0.39	0.05
	10^4	140.41	0.47		
10^5	10^3	149.30	0.53	4.96	0.49
	10^4	ne	3.06		
10^6	10^3	ne	4.11	66.62	5.07
	10^4	ne	69.90		

Note: Computer with a duo-core, 2.1 GHz processor, 16 GB of RAM, running Ubuntu 16.04.

An effort was done to improve this aspect. Table 1 compares the computing times of the two main input functions in *ape* when reading FASTA files with DNA sequences from 10 Mb to 10 Gb and phylogenies with 10 000 to one million tips. The comparisons are between *ape* 5.0 and two older versions that lacked the improvements available in this recent version.

With the increasing availability of parallel computer architectures, R has provided support for parallel computation for some years. *ape* benefits from this to run parallel bootstraping in its function `boot.phylo`: this is done by using the new option mc.cores and specifying the number of cores available or to be used. Furthermore, many basic functions of *ape* have been improved, sometimes by relying on C++ code thanks to the *Rcpp* package (Eddelbuettel, 2013).

Our experience when teaching the use of *ape* has revealed practical difficulties that users sometimes meet and potentially hampers efficient analyses. We have thus developed a set of functions to find data (trees and sequences) files on the user's machine. The function `Xplorefiles` returns a list with all data files searched by default from the HOME directory and identified by the file extensions. The list of these extensions can be modified in a user-friendly way from R. The function `Xplor` performs the same task but outputs its results in the default Web browser with the possibility to display the results in two tabs: sorted either by file types or by directory. In both cases, the full paths to the files are displayed which help the user to read them subsequently in R or even to open them directly in the browser if supported.

Another common difficulty is to compare two trees, for instance estimated with two different methods. The new function `comparePhylo` compares two trees taking into account whether they are rooted or unrooted by comparing their clades or their bipartitions, respectively. A graphical display of the differences is optional. Below is a small example comparing the trees `mytree_1` and `mytree_2` (Fig. 1):

```
> comparePhylo(mytree_1, mytree_2, plot = TRUE)
=> Comparing mytree_1 with mytree_2.
Both trees have the same number of tips: 5.
Both trees have the same tip labels.
Both trees have the same number of nodes: 3.
Both trees are unrooted.
Both trees are not ultrametric.
1 split in common.
```

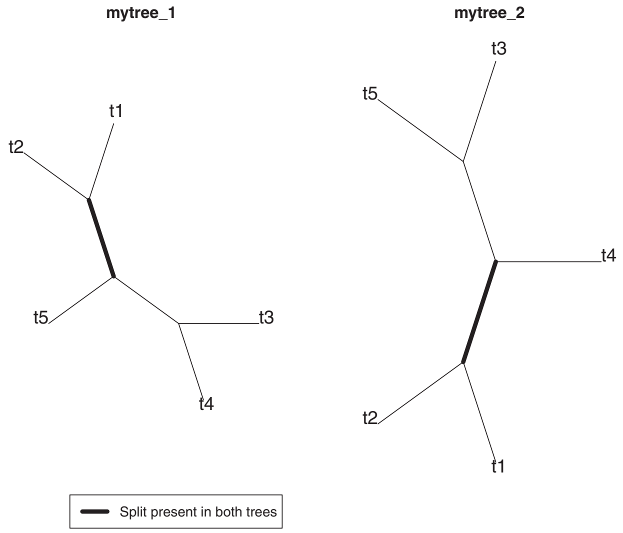

Fig. 1. Comparison of two unrooted trees with *ape* 5.0

A strength of R (probably one of its most important) is the ability to match different datasets using labels (names, rownames and colnames). *ape* uses this feature to match trees, sequences and other data. In this last release, some effort has been done in order to improve the management of labels by the user. The examples below illustrate simply the use of four of these functions with a short vector of three labels x showing how to abbreviate them, make them unique, or building a table with different taxonomic levels:

```
> x <- c("Panthera leo leo", "Panthera tigris",
    + "Panthera pardus")
> stripLabel(x)
[1] "Panthera leo" "Panthera tigris"
[3] "Panthera pardus"
> (y <- stripLabel(x, species = TRUE))
[1] "Panthera" "Panthera" "Panthera"
> makeLabel(y)
[1] "Panthera1" "Panthera2" "Panthera3"
> abbreviateGenus(x)
[1] "P. leo leo" "P. tigris" "P. pardus"
> label2table(x)
    genus species subspecies
1 Panthera leo        leo
2 Panthera tigris     <NA>
3 Panthera pardus     <NA>
```

With the foreseen continuous increase in DNA data quantity from many organisms, we hope that *ape* will continue to be a useful software for manipulating and analyzing these data as well as interpreting them to study biodiversity and evolution.

Acknowledgements

We thank many users for their feedback, and Julien Clavel and one anonymous reviewer for their comments. This is publication ISEM 2018-113.

Conflict of Interest: none declared.

References

Cardona,G. *et al*. (2008) Extended Newick: it is time for a standard representation of phylogenetic networks. *BMC Bioinformatics*, **9**, 532.

Eddelbuettel,D. (2013). *Seamless R and C++ Integration with Rcpp*. Springer, New York.

Huber,W. *et al*. (2015) Orchestrating high-throughput genomic analysis with Bioconductor. *Nat. Methods*, **12**, 115–121.

Paradis,E. *et al*. (2004) APE: analyses of phylogenetics and evolution in R language. *Bioinformatics*, **20**, 289–290.

R Core Team (2017) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.