rtrees: an R package to assemble phylogenetic trees from megatrees

Daijiang Li[1,2]*

17 October, 2022

[1]Department of Biological Sciences, Louisiana State University, Baton Rouge, LA, USA

[2]Center for Computation & Technology, Louisiana State University, Baton Rouge, LA, USA

* **Corresponding author**, email: daijianglee@gmail.com; 125 Life Science Building, Baton Rouge, LA 70803

Running headline: Tool to assemble phylogenetic trees

Number of words: 3985; **Number of Tables**: 1; **Number of Figures**: 1

Abstract: Despite the increasingly available phylogenetic hypotheses for multiple taxonomic groups, most of them do not include all species. In phylogenetic ecology, there is still strong demand to have phylogenies with all species in a study included. The existing software tools to graft species to backbone megatrees, however, are mostly limited to a specific taxonomic group such as plants or fishes. Here, I introduce a new user-friendly R package rtrees that can assemble phylogenies from existing or user-provided megatrees. For most common taxonomic groups, users can only provide a vector of species scientific names to get a phylogeny or a set of posterior phylogenies from megatrees. It is my hope that rtrees can provide an easy, flexible, and reliable way to assemble phylogenies from megatrees, facilitating the progress of phylogenetic ecology.

Keywords: Pruned phylogeny, Supertree, Phylogenetics, Phylogenetic ecology, Community ecology

Introduction

Phylogenetic trees represent hypotheses about the evolutionary history of species, providing essential context for us to understand a wide range of ecological and evolutionary questions such as trait evolution, species interactions, and community assembly (Faith 1992, Webb et al. 2002, Cavender-Bares et al. 2009, Baum and Smith 2012). With the increasingly available genetic and fossil data as well as the development of theories and software tools, established backbone phylogenies are now available for multiple taxonomic groups (e.g., Jetz et al. 2012, Hinchliff et al. 2015, Tonini et al. 2016, Faurby et al. 2018, Jetz and Pyron 2018, Rabosky et al. 2018, Smith and Brown 2018, Stein et al. 2018, Upham et al. 2019). The increasing availability of phylogenies has advanced multiple fields, with phylogenetic ecology being at the top of the list (Webb et al. 2002, Cavender-Bares et al. 2009, Maclvor et al. 2016, Swenson 2019).

Comprehensive phylogenies with as many of the target species to be included as possible are needed for studies in phylogenetic ecology. However, it is still common that only a fraction of the target species can be found in the available phylogenies for many taxonomic groups becaus of the lack of sequence data. This situation leaves two options for ecologist. The first one is to collaborate with phylogenetists to generate their own phylogenies based on sequence data so that all target species will be included. Such phylogenies were referred as purpose-build phylogenies (Li et al. 2019). This option normally requires a large amount of effort to sample sequence data and then to assemble a phylogeny using established methods, which requires specific expertise to being applied appropriately. The second option is to derive their phylogenies from existing large synthetic phylogenies by grafting missing species onto the synthetic phylogenies using taxonomic ranks. Such phylogenies were referred as synthesis-based phylogeny (Li et al. 2019), which have been demonstrated to give similar results as those based on purpose-build phylogenies for most questions asked in phylogenetic ecology such as calculating phylogenetic diversity and estimating phylogenetic signal of traits (Swenson 2009, Cadotte 2015, Li et al. 2019).

Several computational tools exist to derive sythesis-based phylogenies. The oldest and most widely used one is phylomatic (Webb and Donoghue 2005). Since published in 2005, phylomatic has been cited more than 1,000 times and contributed significantly to the development of the

field of phylogenetic ecology. Phylomatic was written with C (standalone program) or GNU Awk (online program) and is fast. However, the warning/error messages sometimes are hard for users to understand. Furthermore, the grafting process within phylomatic depends on the node information of the mega-trees; however, phylomatic did not pre-process the mega-trees (at least for user provided ones) and thus only be able to graft species with congeneric in the mega-trees. Phylomatic provided a standalone and an online version. To use its standalone version, users need to have some basic knowledge about the command line (i.e., terminal). Its online service is easier to use. However, it limits the number of target species to be no more than 5,000 and is likely to be discontinoued in the future (it was out of service for a while during 2021). Another recent similar tool is the set of S.PhyloMaker, V.PhyloMaker, and V.PhyloMaker2, a serial of R packages developed by the same group to derive phylogenies for vascular plant species (Jin and Qian 2019). These packages work well but are limited to vascular plants and their package structures and documentations can be improved by following the best practices of R package development (Wickham 2015). For example, a simple check of the V.PhyloMaker package gave 2 warnings and 4 notes. Both phylomatic and the V.PhyloMaker set of R packages require the users to provide the taxonomic classification (genus and family) of the target species. Another R package FishPhyloMaker was developed recently to derive sythesis-based phylogenies for finned-ray fishes (Nakamura et al. 2021) based on the fish tree of life megatree (Rabosky et al. 2018). Users can simply provide a list of species names and FishPhyloMaker will retrive the taxonomic classification information; when such information cannot be found, users are asked to enter it manually. It also will retrive such information for the tips in the megatree that are in the same genus (if congeneric species exist in the megatree), family (if no congeneric species in the megatree), or order (if no co-family species in the megatree) of the target species with every call of the function. Such a design requires internet access and can be slow.

What is missing from the tool box of phylogenetic ecologists is a user friendly program that can derive synthesis-based phylogenies for most common taxonomic groups with existing megatrees by taking a species list. If a large set of posterior phylogenies exist for some taxonomic groups, the tool should be able to derive phylogenies from a small number of randomly selected megatrees (e.g., 100) so that uncertainties can be accounted for in the downstream analyses. To fill this gap, I developed an R package named rtrees. With rtrees, users only need to provide a

species list to derive phylogeny or phylogenies for taxonomic groups with existing established megatrees, which have been processed and hosted in a separate R data package megatrees. Users can also provide their own megatrees if needed. For species list with over 200 species, a progress bar will show to indicate the percentage of jobs that have been finished. I also provided a shinny app for users who want to derive phylogenies online when the number of missing species is less than 1,000.

Package installation

```
install.packages('rtrees', repos = 'https://daijiang.r-universe.dev')
```

The code above will also install the data package megatrees, which hosts a collection of existing synthetic megatrees for amphibians, birds, fishes, mammals, plants, reptiles, and sharks (see Table 1 for details).

Usage scenarios

rtrees can be used in the following exemplary scenarios. In all scenarios, we can use the function rtrees::get_tree() to get the phylogenies. See the examples section for details about its usage.

1. We have a list of species, and we want to derive a phylogeny of those species from existing megatrees (i.e., synthetic phylogenies) for our analyses such as estimating phylogenetic signal of species traits, or calculating phylogenetic diversity and investigating phylogenetic structures of communities. For these kinds of analyses, a phylogeny derived from synthetic megatree provides robust results (Li et al. 2019).
2. We have a phylogeny based on sequence data (i.e., a purpose-build phylogeny) and we want to insert more species in the phylogeny, potentially as polytomies with their congeneric or co-family species in the phylogeny because of the lack of sequence data.

3. We have a phylogeny for the species pool. However, we have species that cannot be identifed at species level with certainty (e.g., *Carex spp*). Such morphological species can be binded into the phylogeny using rtrees so that we don't need to throw them away for the downstream analysis.
4. We can find a set of existing posterior phylogenies that include all or most of our species. However, there are thousands of such posterior phylogenies in the datasets (e.g., phylogenies provided by VertLife https://vertlife.org/). For our analyses, we only need a smaller number of randomly selected posterior phylogenies to account for the uncertainty of phylogenetic hypotheses. Previous studies suggested that 100 randomly selected phylogenies is generally enough to capture the uncertainties (Li et al. 2018, Nakagawa and De Villemereuil 2019, Upham et al. 2019). R data package megatrees can save us time to repeat this download-and-subset process by providing 100 randomly selected posterior phylogenies. R package rtrees will use these 100 randomly selected phylogenies to derive 100 phylogenies for the species list we have.

Shinny app for a small number of species

I have also developed a shinny app (https://djli.shinyapps.io/rtrees_shiny/) to get phylogenies when the number of missing species is small (< 1,000). The main usage scenario of this shinny app is when we want to quickly get a phylogeny or phylogenies for a small number of species without using R. Another main reason to have this limit of 1,000 is because I cannot afford the paid plans provided by the shinny hosting site. When the number of missing species is over 1,000, we should use the R package instead.

Package Strcture

Classification information

The taxonomic classification information (e.g., genus and family of each species) is critical for the pre-process of megatrees (see below) and to determine where a new species should

be grafted onto a megatree. Therefore, rtrees provides such information in the R object rtrees::classifications for common taxonomic groups. In the current version of rtrees (v0.0.2), the object rtrees::classifications includes 24,222 unique genus of plants, 4,833 unique genus of fishes, 2,508 unique genus of birds, 1,419 unique genus of mammals, 1,237 unique genus of reptiles, 543 unique genus of amphibians, and 198 unique genus of sharks, rays, and chimaeras. I did not include classification information above the family level (e.g., order) because grafting species above family level may bring too much uncertainties.

For plants, I extracted genus and family information from multiple sources, including the Plant List (superseded by the World Flora Online), the World Flora Online, the Plant of World Online, and Catelogue of Life 2019. When different sources give different family information for the same genus, I used the information provided by the Plant of World Online. For fish, I used the taxonomic information provided by the fish tree of life (Rabosky et al. 2018). Jetz et al. (2012) build the bird phylogenies based on the Handbook of the Birds of the World (HBW) and BirdLife International digital checklist version 3 and later updated the taxonomy based on version 5. I downloaded the bird taxonomy from Birdtree.org. Mammal taxonomy information are from two sources: PHYLACINE 1.2 (Faurby et al. 2018) and the mammal diversity database of VertLife (Upham et al. 2019). For genus with different family information from these two sources, I used the information provided by the VertLife. Taxonomy information for amphibians, reptiles, sharks, rays, and chimaeras were all provided by the Vertlife (https://vertlife.org/).

Sources and preparation of megatrees

Sources of exisiting megatrees for different taxonomic groups were described in Table 1. For taxonomic groups with multiple posterior phylogenies, a subset of phylogenies (100 for taxonomic groups except fish, see the column of # of trees in Table 1) were randomly selected. For most analyses, 50-100 randomly selected posterior phylogenies are enough to account for the uncertainties in phylogenetic spaces (Li et al. 2018, Nakagawa and De Villemereuil 2019, Upham et al. 2019). All megatrees were stored in the megatrees R data package with class of phylo or multiPhylo, the most common data structures of phylogenies used in R.

Each megatree was processed so that the most recent common ancestors (MRCA) of all the

Table 1. Brief information about the megatrees included in the megatrees package, which will be installed automatically when rtrees was installed.

Taxon	# of species	# of trees	R object	Reference
Amphibian	7,238	100	tree_amphibian_n100	Jetz and Pyron 2018
Bird	9,993	100	tree_bird_n100	Jetz et al. 2012
Fish	11,638	1	tree_fish_12k	Rabosky et al. 2018
	31,516	50	tree_fish_32k_n50	Rabosky et al. 2018
Mammal	5,831	100	tree_mammal_n100_phylacine	Faurby et al. 2018
	5,911	100	tree_mammal_n100_vertlife	Upham et al. 2019
Plant	74,531	1	tree_plant_otl	Smith and Brown 2018
Reptile (Squamate)	9,755	100	tree_reptile_n100	Tonini et al. 2016
Shark, Ray, and Chimaera	1,192	100	tree_shark_ray_n100	Stein et al. 2018

genus and family in the megatrees are determined using the function rtrees::add_root_info(). If a genus or a family is not monophyletic, I used the most inclusive MRCA for that genus or family. The MRCA information of each megatree was saved as an extra component named as genus_family_root in the corresponding R object (see Table 1; e.g., tree_plant_otl$genus_family_root). The function rtrees::add_root_info() will also be used by rtrees::get_tree() to process user-provided megatrees.

Species names processing

R package rtrees does not provide functions to standardize taxonomic names because exisiting packages such as taxize already provide such features. Users should use such exisiting tools to standardize their species names first, ideally using the same taxonomy backbone as the corresponding megatrees described in Table 1. When users provide a list of standardized species without information about genus and family (can be a character vector or a data frame with one column named as 'species'), function rtrees::get_tree() will automatically call function rtrees::sp_list_df() to use the classification information described above to extract the genus and family information needed for binding missing species to the megatrees if the taxonomic group is one of those in Table 1. Note that if all genus of the species list are already in the megatrees, no classification information will be needed to finish the binding process. Users can also pass prepared classification information to rtrees::get_tree(). To do so, the input data frame should have at least three columns: species, genus, and family. Two extra optional

columns (close_sp and close_genus) can also be included in the input data frame to specify where the target species should be grafted into the megatrees. If users provided the classification and/or location information, rtrees will honor the user provided information (see examples below).

The binding process

Once the megatrees were processed and the classification information of species were ready, the binding process begins. If all species are already present in the megatrees, no binding will be needed and a prunned phylogeny will be returned. Otherwise, for species that are missing from the megatrees, rtrees will first look for congeneric species in the megatrees. If no congeneric species was found in the megatrees, rtrees will then look for co-family species in the megatrees. If neither congeneric nor co-family species were found, the target species will be skipped and will be included in the output message with other skipped target species. If either congeneric or co-family species were found in the megatrees, users have three options to bind the target species into the megatrees by setting the scenario argument within rtrees::get_tree():

- The default way is to bind the missing target species as a polytomy at the basal node of the MRCA of the genus or family in the megatrees (scenario = "at_basal_node"); if the megatrees have only one species in the that genus or family, then the missing target species will be inserted to the half of this only species' branch length.
- If users set scenario to be "random_below_basal", a randomly selected node within the genus or family will be used to bind the missing target species; the probability of a node been selected is proportional to its branch length. Because of the randomness involved with this option, users may want to run this option multiple time (e.g., 50-100) to generate a set of phylogenies.
- If users set scenario to be "at_or_above_basal", then the missing target species will be binded to the basal node of the same genus or to a new node above the basal node of the family if no congeneric species were found in the megatrees. If the age of the family's basal node is less than 2/3 of the node above it (root node of the family), the missing target species will be binded so that its age will be 2/3 of the root node of the family. Otherwise,

the missing target species will be inserted into the middle point of the basal node and the root node of the family. When using the phylogenies derived with this option to calculate community phylogenetic diversity, the results may be inflated (the most among these three scenarios).

By default, if the number of missing target species is over 200, a progress bar will be shown in the console. Once the binding process is finished, the megatrees will be pruned to only keep the target species. The generated phylogeny will then be ladderized with R function ape::ladderize(). When there is only one megatree used, the generated phylogeny will have a class of 'phylo'; when multiple posterior megatrees were used, the generated phylogenies will have a class of 'multiPhylo'. When show_grafted is set to TRUE (default is FALSE) within rtrees::get_tree(), binded species will be indicated with trailing * or ** in the tip labels of the generated phylogeny, indicating that a species was binded at the genus and family level, respectively. If such information is important for downstream analyses, users can extract such information as its own data frame using rtrees::get_graft_status(). Users can use rtrees::rm_stars() to remove all trailing stars from the tip labels of the generated phylogeny.

Main functions and example applications

In this section, I provide exemplary code to deal with the different situations described in the Usage scenarios section above. The main function to use is rtrees::get_tree(). Users can type ?rtrees::get_tree in the R console to see details.

For the first scenario described in the Usage scenarios section above (i.e., to get a phylogeny for a list of species using an existing synthetic megatree), we can use the following code:

```
# create a species list
species <- c('Meliosma laui', 'Acer cordatum', 'Fraxinus mandshurica',
             'Ormosia pinnata', 'Aglaia dasyclada', 'Sphagnum_subnitens',
             'Stephanomeria_cichoriacea', 'Taraxacum_schroeterianum',
             'Humiria_balsamifera', 'Salix_cinerea', 'Floerkea_proserpinacoides')
```

```
# get a phylogeny
sp_tree <- rtrees::get_tree(sp_list = species, taxon = 'plant')
```

```
##
## 5 species added at genus level (*)

## 1 species added at family level (**)

## 1 species have no co-family species in the mega-tree, skipped
## (if you know their family, prepare and edit species list with `rtrees::sp_list_df()` may help):
## Sphagnum_subnitens
```

In the code chunk above, the vector of species names can have either space or underscore in the names; spaces will be replaced by underscores internally within rtrees::get_tree(). If no megatree was set by the tree = argument, then the taxon = argument must be one of the supported taxonomic groups ("amphibian", "bird", "fish", "mammal", "plant", "reptile", and "shark_ray"). Function rtrees::get_tree() will print out messages about the number of species been grafted at genus and family level as well as the number of species been skipped if neither congeneric nor co-family species were found in the megatree. In the example above, the skipped species is a moss and the megatree does not have any moss species from the Sphagnaceae family. When the argument show_grafted of rtrees::get_tree() is set to 'TRUE' (default is 'FALSE'), the tip labels of the generated phylogeny will have a trailing * if it is grafted at genus level or ** if it is grafted at family level. No matter whether show_grafted is 'TRUE' or 'FALSE', the grafting information was saved along with the phylogeny and can be extracted with the following code:

```
# or use rtrees::get_graft_status()
sp_tree$graft_status
```

```
## # A tibble: 11 x 3
##    tip_label            species              status
##    <chr>                <chr>                <chr>
```

```
##  1 Taraxacum_schroeterianum  Taraxacum_schroeterianum  exisiting species in the megatree
##  2 Stephanomeria_cichoriacea Stephanomeria_cichoriacea exisiting species in the megatree
##  3 Fraxinus_mandshurica      Fraxinus_mandshurica      grafted at genus level
##  4 Ormosia_pinnata           Ormosia_pinnata           grafted at genus level
##  5 Salix_cinerea             Salix_cinerea             exisiting species in the megatree
##  6 Humiria_balsamifera       Humiria_balsamifera       exisiting species in the megatree
##  7 Floerkea_proserpinacoides Floerkea_proserpinacoides grafted at family level
##  8 Aglaia_dasyclada          Aglaia_dasyclada          grafted at genus level
##  9 Acer_cordatum             Acer_cordatum             grafted at genus level
## 10 Meliosma_laui             Meliosma_laui             grafted at genus level
## 11 <NA>                      Sphagnum_subnitens        skipped as no co-family in the m~
```

When users already have a phylogeny for most of their species (i.e., the second and third scenarios described in the Usage scenarios section above), we can use the same code as above, with the argument tree_by_user = TRUE. And here is an example using the phylogeny generated above as a pretended megatree that we already have.

```
more_sp_to_add = c('Ormosia_sp.', 'Fraxinus_americana')
new_species = c(species, more_sp_to_add)
sp_tree_2 = rtrees::get_tree(sp_list = new_species, tree = sp_tree,
                             taxon = 'plant', tree_by_user = TRUE)
```

```
## Not all genus can be found in the phylogeny.

##
## 2 species added at genus level (*)

## 1 species have no co-family species in the mega-tree, skipped
## (if you know their family, prepare and edit species list with `rtrees::sp_list_df()` may help):
## Sphagnum_subnitens
```

In the code above, as there is a genus (Sphagnum) not included in the user provided phylogeny, we need to specify the taxon argument to extract the correct classification information; note that this requires the taxonomic group is one of those supported by rtrees (Table 1). However, if all genus of the species list are already in the user provided phylogeny, then we can ignore the taxon argument:

```
# remove Sphagnum_subnitens so that all genus are in the megatree
new_species_all_in = setdiff(new_species, 'Sphagnum_subnitens')
sp_tree_3 = rtrees::get_tree(sp_list = new_species_all_in, tree = sp_tree,
                             tree_by_user = TRUE)
```

```
##
## 2 species added at genus level (*)
```

The function rtrees::get_tree() can also work with a set of posterior megatrees (the fourth scenario in the Usage scenarios section) with the option to use parallel computing for the whole process. The default number of cores to be used will be the available number of cores minus 2 (so that users can still perform other tasks on their computers at the same time). The output will be a set of generated phylogenies with class 'multiPhylo'; the number of derived phylogenies will be the same as the input megatrees. For this scenario, we can use exactly the same code described above.

```
# bird species
bird_species =  c('Brachypteryx_major', 'Asthenes_perijana', 'Ciridops_anna',
                  'Leiothlypis_ruficapilla', 'Reinwardtoena_reinwardti',
                  'Garrulax_caerulatus', 'Buteo_rufofuscus', 'Sylvia_mystacea',
                  'Telophorus_viridis', 'Trachyphonus_darnaudii')
sp_tree_4 = rtrees::get_tree(sp_list = bird_species, taxon = 'bird')
```

```
##
## 3 species added at genus level (*)
```

2 species added at family level (**)

```
sp_tree_4
```

100 phylogenetic trees

Performance

The R package rtrees is a user friendly tool to assemble phylogenies from existing or user provided megatrees despite that it is not as fast as phylomatic (Webb and Donoghue 2005), which was written in C. However, phylomatic can give very uninformative error messages in most cases that I have used it. For example, messages below would make it really hard to debug:

Error: Program 'phylomatic' terminated by SIGNAL (Segmentation fault: 11)

bad CPU type in executable: phylomatic

I tried to compare the speed of rtrees and phylomatic but was not able to run phylomatic with the large megatrees prepared here after multiple attempts. On the other hand, I was able to compare the performance of rtrees and V.PhyloMaker. In most cases, rtrees is at least two times faster than V.PhyloMaker (Fig. 1). For example, with 50 missing species to bind, the average time used by rtrees is 0.914 second while V.PhyloMaker took 13.6 seconds on average; with 5,000 missing species to bind, rtrees used 80.9 seconds on average while V.PhyloMaker used 187 seconds. In general, the time used by rtrees and V.PhyloMaker increased 15.6 and 34.8 seconds on average with 1,000 more missing species to be grafted, respectively. All tests were conducted on a 14' Macbook Pro with M1 pro chip.

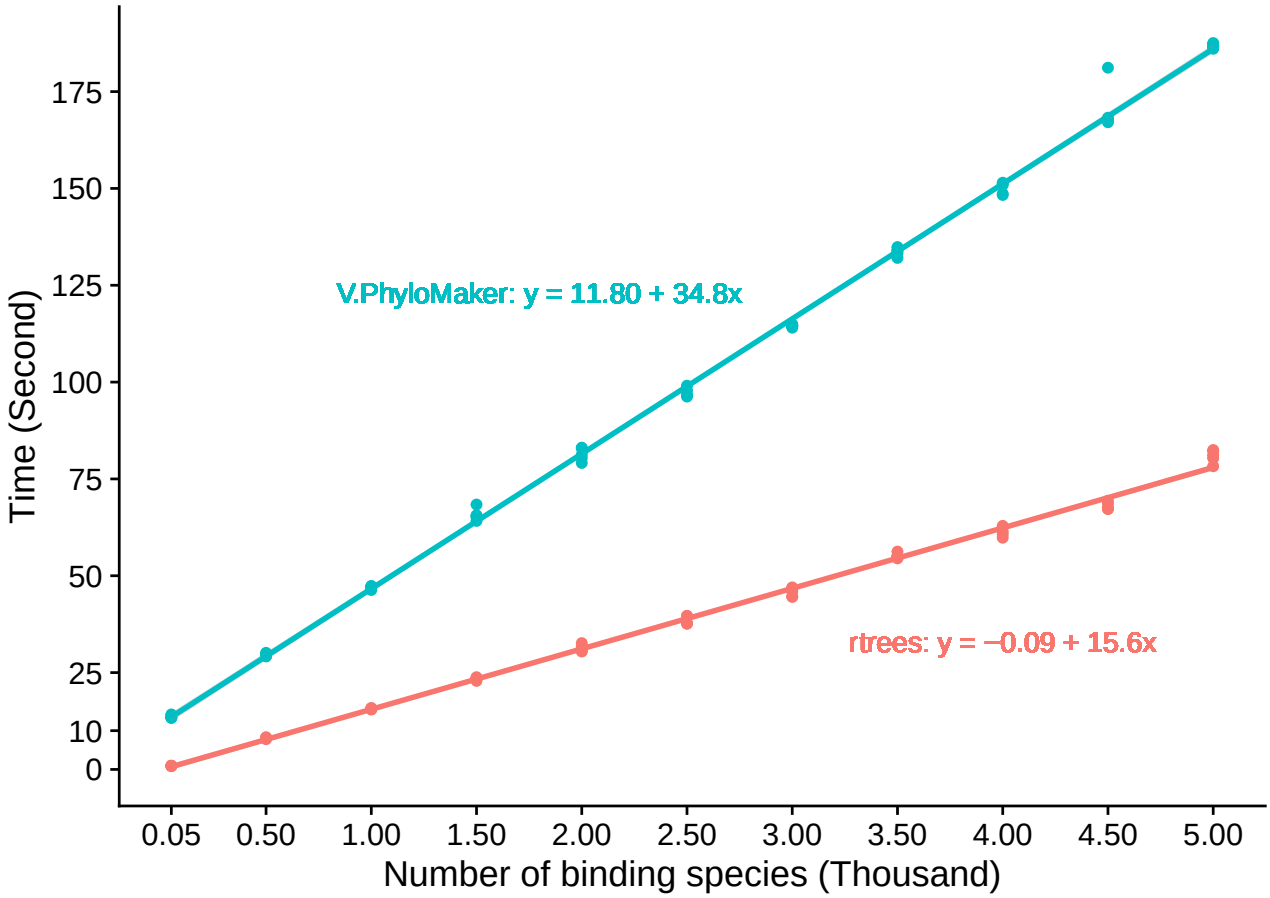

Figure 1. Performance comparisons between rtrees and V.PhyloMaker. For each test, five replications were conducted using both the rtrees and V.PhyloMaker R package.

I did not compare the performance between rtrees and FishPhyloMaker here because the later' design to process the fish megatree (e.g., find MRCA for clades) every time whenever we use it makes it really slow. For example, to bind 500 missing fish species, FishPhyloMaker took almost 8 minutes while rtrees used only 4 seconds.

Discussion

Based on Fig. 1, the time used by rtrees (and other R packages such as V.PhyloMaker and FishPhyloMaker) increased linearly with the number of missing species. This is because of the sequential structure of the R phylo object (Paradis et al. 2004), in which all tips and nodes of a phylogeny are labeled with continoued integers. When a new node or tip needs

to be inserted, all the numbers after the binding location need to be pushed one place after. Therefore, the binding of tips in an R phylo object is a sequential process, which cannot be conducted with parallel computing to take advantage of the multiple CPU cores a regular computer has nowadays. It is possible to speed it up by converting some core R code into C++. However, it takes a lot of effort to do this right while I don't have extensive knowledge of C++. Another potential resolution is to represent phylogenies in a non-sequential structure unlike phylo objects in R. One example is the mantree package (Bennett et al. 2017), which developed such a structure to perform tree manipulations. Because of the non-sequential structure (i.e., no continous labeling of nodes and tips), it is possible to use parallel computing to process phylogenies. However, such a structure has not yet been adapted widely by the R community. Since the publication of the original paper (Bennett et al. 2017) in 2017, it has been cited for only 17 times by October 2022 based on Google Scholar (by contrast, R package ape has been cited over 11,100 times since 2004). Therefore, the infrastructure to support such a non-sequential structure in R is still missing, making it hard to use it here in rtrees. Despite the limitations described above, rtrees is reasonably fast and user friendly with informative messages. The R package rtrees is also the only R package that can generate phylogenies for multiple taxonomic groups (Table 1) without much effort from the user side. Other packages such as V.PhyloMaker and phylomatic can do this in theory (as claimed by authors in their papers) (Webb and Donoghue 2005, Jin and Qian 2019); however, users need to figure out how to do it by themselves.

With the recent advances in phylogenetics of multiple taxonomic groups, more megatrees will be available in the near future (e.g., Lepidoptera (Kawahara et al. 2019)). It is relatively easy to include more megatrees beyond those described in Table 1 as the R package rtrees was designed with expandibility in mind. We only need to do two things to include a new megatree. First, the new megatree will be processed with the function rtrees::add_root_info() and will be stored in the R data package megatrees, which is a dependency of rtrees. Second, the classification information (genus and family) will be saved within rtrees if it is a new taxonomic group. No further change will be needed for other components of rtrees.

It is my hope that rtrees will make it much easier to derive phylogenies from existing megatrees for all common taxonomic groups. Such synthesis-based phylogenies are reliable for most

ecological questions such as calculating phylogenetic diversity and estimating phylogenetic signals (Li et al. 2019). Note that such phylogenies may not be suitable for evolutionary studies such as estimating diversification rates if a large proportion of missing species were included in the derived phylogenies. Therefore, it is also my hope that rtrees will facilitate research in phylogenetic ecology. I am committed to maintain and update rtrees in the foreseeable future. Since rtrees is an open source software, others are more than welcome to contribute by submitting pull requests or openning issues to the GitHub repository (https://github.com/daijiang/rtrees).

Data and code availability

No empirical data were used in this paper. The code used for the performance tests can be found in the GitHub repository of this paper (https://github.com/daijiang/rtrees_ms).

Acknowledgement

This study was partly supported by NSF grant DEB #2213567. I thank Harroop Bedi for his contribution on developing the prototype of the rtrees shinny app through the REU program at the Center of Computation & Techology at the Louisiana State University supported by NSF grant OAC #1852454.

References

Baum, D. A., and S. D. Smith. 2012. Tree thinking: An introduction to phylogenetic biology. Roberts; Co., Greenwood Village, CO.

Bennett, D. J., M. D. Sutton, and S. T. Turvey. 2017. Treeman: An r package for efficient and intuitive manipulation of phylogenetic trees. BMC research notes 10:1–10.

Cadotte, M. W. 2015. Phylogenetic diversity–ecosystem function relationships are insensitive to phylogenetic edge lengths. Functional Ecology 29:718–723.

Cavender-Bares, J., K. H. Kozak, P. V. Fine, and S. W. Kembel. 2009. The merging of community ecology and phylogenetic biology. Ecology letters 12:693–715.

Faith, D. P. 1992. Conservation evaluation and phylogenetic diversity. Biological conservation 61:1–10.

Faurby, S., M. Davis, R. Ø. Pedersen, S. D. Schowanek, A. Antonelli, and J.-C. Svenning. 2018. PHYLACINE 1.2: The phylogenetic atlas of mammal macroecology. Ecology 99:2626.

Hinchliff, C. E., S. A. Smith, J. F. Allman, J. G. Burleigh, R. Chaudhary, L. M. Coghill, K. A. Crandall, J. Deng, B. T. Drew, R. Gazis, and others. 2015. Synthesis of phylogeny and taxonomy into a comprehensive tree of life. Proceedings of the National Academy of Sciences 112:12764–12769.

Jetz, W., and R. A. Pyron. 2018. The interplay of past diversification and evolutionary isolation with present imperilment across the amphibian tree of life. Nature ecology & evolution 2:850–858.

Jetz, W., G. Thomas, J. Joy, K. Hartmann, and A. Mooers. 2012. The global diversity of birds in space and time. Nature 491:444–448.

Jin, Y., and H. Qian. 2019. V. PhyloMaker: An r package that can generate very large phylogenies for vascular plants. Ecography 42:1353–1359.

Kawahara, A. Y., D. Plotkin, M. Espeland, K. Meusemann, E. F. Toussaint, A. Donath, F. Gimnich, P. B. Frandsen, A. Zwick, M. Dos Reis, and others. 2019. Phylogenomics reveals the evolutionary timing and pattern of butterflies and moths. Proceedings of the National Academy of Sciences 116:22657–22663.

Li, D., W. B. Monahan, and B. Baiser. 2018. Species richness and phylogenetic diversity of native and non-native species respond differently to area and environmental factors. Diversity and Distributions 24:853–864.

Li, D., L. Trotta, H. E. Marx, J. M. Allen, M. Sun, D. E. Soltis, P. S. Soltis, R. P. Guralnick, and B. Baiser. 2019. For common community phylogenetic analyses, go ahead and use synthesis phylogenies. Ecology 100:e02788.

Maclvor, J. S., J. S. Macivor, M. W. Cadotte, S. W. Livingstone, J. T. Lundholm, and S.-L. E. Yasui. 2016. Phylogenetic ecology and the greening of cities. Journal of Applied Ecology:1470–1476.

Nakagawa, S., and P. De Villemereuil. 2019. A general method for simultaneously accounting for phylogenetic and species sampling uncertainty via rubin's rules in comparative analysis. Systematic Biology 68:632–641.

Nakamura, G., A. Richter, and B. E. Soares. 2021. FishPhyloMaker: An r package to generate phylogenies for ray-finned fishes. Ecological Informatics 66:101481.

Paradis, E., J. Claude, and K. Strimmer. 2004. APE: Analyses of phylogenetics and evolution in r language. Bioinformatics 20:289–290.

Rabosky, D. L., J. Chang, P. F. Cowman, L. Sallan, M. Friedman, K. Kaschner, C. Garilao, T. J. Near, M. Coll, M. E. Alfaro, and others. 2018. An inverse latitudinal gradient in speciation rate for marine fishes. Nature 559:392–395.

Smith, S. A., and J. W. Brown. 2018. Constructing a broadly inclusive seed plant phylogeny. American journal of botany 105:302–314.

Stein, R. W., C. G. Mull, T. S. Kuhn, N. C. Aschliman, L. N. Davidson, J. B. Joy, G. J. Smith, N. K. Dulvy, and A. O. Mooers. 2018. Global priorities for conserving the evolutionary history of sharks, rays and chimaeras. Nature ecology & evolution 2:288–298.

Swenson, N. G. 2009. Phylogenetic resolution and quantifying the phylogenetic diversity and dispersion of communities. PloS one 4:e4390.

Swenson, N. G. 2019. Phylogenetic ecology: A history, critique, and remodeling. University of Chicago Press.

Tonini, J. F. R., K. H. Beard, R. B. Ferreira, W. Jetz, and R. A. Pyron. 2016. Fully-sampled phylogenies of squamates reveal evolutionary patterns in threat status. Biological Conservation 204:23–31.

Upham, N. S., J. A. Esselstyn, and W. Jetz. 2019. Inferring the mammal tree: Species-level sets of phylogenies for questions in ecology, evolution, and conservation. PLoS biology 17:e3000494.

Webb, C. O., D. D. Ackerly, M. A. McPeek, and M. J. Donoghue. 2002. Phylogenies and community ecology. Annual review of ecology and systematics 33:475–505.

Webb, C. O., and M. J. Donoghue. 2005. Phylomatic: Tree assembly for applied phylogenetics. Molecular Ecology Resources 5:181–183.

Wickham, H. 2015. R packages: Organize, test, document, and share your code. " O'Reilly Media, Inc.".