Treeio: An R Package for Phylogenetic Tree Input and Output with Richly Annotated and Associated Data

Li-Gen Wang,¹ Tommy Tsan-Yuk Lam,² Shuangbin Xu,¹ Zehan Dai,¹ Lang Zhou,¹ Tingze Feng,¹ Pingfan Guo,¹ Casey W. Dunn,³ Bradley R. Jones,⁴ Tyler Bradley,⁵ Huachen Zhu,^{2,6,7} Yi Guan,^{2,6,7} Yong Jiang,⁸ and Guangchuang Yu^{*,1,8}

¹Department of Bioinformatics, School of Basic Medical Sciences, Southern Medical University, Guangzhou, Guangdong, China

²State Key Laboratory of Emerging Infectious Disease and Centre of Influenza Research, School of Public Health, The University of Hong Kong, Hong Kong SAR, China

³Department of Ecology and Evolutionary Biology, Yale University, New Haven, CT

⁴British Columbia Centre for Excellence in HIV/AIDS, Vancouver, Canada

⁵Civil, Architectural, and Environmental Engineering, Drexel University, Philadelphia, PA

⁶Joint Institute of Virology (Shantou University – The University of Hong Kong), Shantou University, Shantou, Guangdong, China

⁷State Key Laboratory of Emerging Infectious Disease (HKU Shenzhen Base), Shenzhen Third People's Hospital, Shenzhen, Guangdong, China

⁸Guangdong Provincial Key Laboratory of Proteomics, School of Basic Medical Sciences, Southern Medical University, Guangzhou, Guangdong, China

*Corresponding author: E-mail: gcyu1@smu.edu.cn.

Associate editor: Sudhir Kumar

Abstract

Phylogenetic trees and data are often stored in incompatible and inconsistent formats. The outputs of software tools that contain trees with analysis findings are often not compatible with each other, making it hard to integrate the results of different analyses in a comparative study. The *treeio* package is designed to connect phylogenetic tree input and output. It supports extracting phylogenetic trees as well as the outputs of commonly used analytical software. It can link external data to phylogenies and merge tree data obtained from different sources, enabling analyses of phylogeny-associated data from different disciplines in an evolutionary context. *Treeio* also supports export of a phylogenetic tree with heterogeneous-associated data to a single tree file, including BEAST compatible NEXUS and jtree formats; these facilitate data sharing as well as file format conversion for downstream analysis. The *treeio* package is designed to work with the *tidytree* and *ggtree* packages. Tree data can be processed using the tidy interface with *tidytree* and visualized by *ggtree*. The *treeio* package is released within the Bioconductor and rOpenSci projects. It is available at <https://www.bioconductor.org/packages/treeio/>.

Key words: data integration, phylogeny, *ggtree*, bioconductor.

Introduction

Along with genomic sequences, the amount of phylogenetic data has rapidly increased. These phylogeny-associated data, including traits, metadata (e.g., host from which the sample was isolated, time and location of sample collection, and so on), experimental and clinical data, as well as analysis findings and evolutionary inferences from commonly used software in this field, need to be linked to the taxon species on the tree for further analyses in a phylogenetic context and comparative studies. Despite the rapid growth in data generation, integrative and comparative analyses that combine different data sources are rare. Integrating phylogenetic data are difficult for several reasons, including incompatibility of file formats and difficulty of linking data to taxon species or branches. To solve these issues, we present an R package, *treeio*, for basic input and output of phylogenetic trees and associated data. The

treeio package includes a number of parser functions to import trees with or without data from different file formats as well as the outputs of commonly used software, utilities to link external data to phylogenies and manipulate tree data, and exporter functions to write a tree with associated data to a variety of file formats (fig. 1). Combining a tree with data in a broadly compatible format can facilitate data sharing and enhance reproducibility and reuse.

Importing Richly Annotated Phylogenetic Data

Software packages capable of evolutionary inference are proliferating, for example, BEAST (Bouckaert et al. 2014) for phylodynamic inferences and CodeML (Yang 2007) for inferring ancestral sequences and selection pressure. However, the outputs of these commonly used software packages are not

FIG. 1. Overview of the *treeio* package and its relations with *tidytree* and *ggtree*. *Treeio* supports parsing a tree with data from a number of file formats and software outputs. A *treedata* object stores a phylogenetic tree with node/branch-associated data. *Treeio* provides several functions to manipulate a tree with data. Users can convert the *treedata* object into a tidy data frame (each row represents a node in the tree and each column represents a variable) and process the tree with data using the tidy interface implemented in *tidytree*. The tree can be extracted from the *treedata* object and exported to a Newick or NEXUS file or can be exported with associated data into a single file (either in the BEAST NEXUS or jtree format). Associated data stored in the *treedata* object can be used to annotate the tree using *ggtree*. In addition, *ggtree* supports a number of tree objects, including *phyloseq* for microbiome data and *obkData* for outbreak data. The *phylo*, *multiPhylo* (*ape* package), *phylo4*, *phylo4d* (*phylobase* package), *phylog* (*ade4* package), *phyloseq* (*phyloseq* package), and *obkData* (*OutbreakTools* package) are tree objects defined by the R community to store tree with or without domain-specific data. All these tree objects as well as hierarchical clustering results (e.g., *hclust* and *dendrogram* objects) are supported by *ggtree*.

standardized and are not necessarily compatible with each other. Extracting analysis results from these inconsistent formats often requires programming expertise. We previously addressed this by implementing a number of parser functions and classes in *ggtree* (Yu et al. 2017). We removed all of the parser functions from *ggtree* and collected them into a new package, *treeio*, with many improvements. Specifically, all of the parser functions now output the same tree object type, *treedata*, making them consistent and robust for downstream manipulation. In addition, the *treeio* package can parse more file formats and software outputs than previously in *ggtree*. Currently, *treeio* can read the following file formats: Newick,

NEXUS (Maddison et al. 1997), New Hampshire eXtended format (NHX), jplace (Matsen et al. 2012), and phylip as well as the data outputs from the following analysis programs: ASTRAL (Zhang et al. 2018), BEAST (Bouckaert et al. 2014), EPA (Berger et al. 2011), HyPhy (Pond et al. 2005), IQ-TREE (Nguyen et al. 2015), MEGA (Kumar et al. 2016), MrBayes (Ronquist and Huelsenbeck 2003), PAML (Yang 2007), PHYLDOG (Boussau et al. 2013), pplacer (Matsen et al. 2010), RAxML (Stamatakis 2014), r8s (Sanderson 2003), and RevBayes (Höhna et al. 2016). After parsing, all of the information is stored in the *treedata* object and the parsed data are mapped to the tree branches and nodes inside the object.

A wide range of heterogeneous data, including phenotypic and experimental data, routinely requires integration with phylogeny in a typical bioinformatics workflow. For instance, mapping functional annotations to a gene tree is useful for comparative genomics studies and mapping geographical positions of the organisms to their phylogenetic trees is helpful for understanding the biogeography of the species. To facilitate data integration, *treeio* provides the *full_join* method (Wickham et al. 2019) that links external data to phylogeny and stores the result in a *treedata* object. The *full_join* method requires a tree object (either *phylo* or *treedata*, *phylo* is defined in the *ape* package to store a phylogenetic tree structure, including tip labels, edge list, and corresponding edge length [optional]) and a data frame as input. The data frame should contain a column with taxon labels or node numbers which will be matched with nodes in the tree. The associated information stored in other columns of the data frame will be linked to corresponding nodes and stored in the *treedata* object. In addition to incorporating data into a tree object with the *full_join* method, users can visualize the data mapped onto the tree using *ggtree* (Yu et al. 2018). Associated data that are supported by *treeio* are also supported by *ggtree*. This creates the possibility to link external data to other tree objects supported by *ggtree* (fig. 1), especially for those that are not designed to store associated data or only for domain-specific data (e.g., microbiome census data stored in a *phyloseq* object). Linking external data to these tree objects is generally not supported at the tree object level, but *ggtree* is able to link external data to them for visualization.

With the ability to parse data from a number of file formats and link external data, *treeio* enables further analyses of phylogenetic data in R. Furthermore, *treeio* can merge multiple *treedata* objects together into one with their node/branch-specific attribute data if the trees have identical topologies. As a result, one attribute (e.g., substitution rate) can be mapped to another attribute (e.g., d_N/d_S or phenotypic trait) of the same node/branch for comparison and further computations. This allows comparison of evolutionary inferences obtained via different models or using different software with identical model and parameters, as well as correlation of estimates to observed data.

Manipulating a Phylogenetic Tree with Data

Before analyzing phylogenetic data, it is important to check data quality, including low sequence quality, errors in sequence assembly or data annotation, sample contamination and alignment error. For example, it is advisable to identify and remove sequences whose genetic divergence and sampling date are incongruent before applying a molecular clock model to virus data (Rambaut et al. 2016). These sequences can be easily removed using the *drop.tip* function. Trees can be large and it can be difficult to explore only the portions of interest. The *tree_subset* function allows us to look at the portions of the tree around a particular clade or tip. It accepts a tree object with the node/tip number or label as input and will take the tip or the clade as a whole and trace back to

specific levels to look at the immediate relatives of the tip/clade. This makes it easy to explore the portions of the tree that are of interest and their closest relatives. The *treeio* package also provides the *root* function to reroot a tree by specifying a node number or an outgroup. All these manipulations maintain all of the branch lengths and related associated data which will be updated according to the change in the tree structure and remains linked to their associated nodes.

Numerical data can be used to rescale the branch lengths of the tree using the *rescale_tree* function. For example, we can use d_N/d_S in place of branch length in a visualization of the tree to inspect evolutionary patterns of genomic divergence. These values can also be projected onto the vertical dimension to create a 2D tree or so-called phenogram using *ggtree* to explore the patterns of trait evolution. The *groupOTU* and *groupClade* functions add grouping information to the tree. The *groupOTU* requires a list of taxon labels as input while *groupClade* accepts a vector of internal node numbers. The *groupOTU* will trace back from specified tips in the input to the most recent common ancestor. Related OTUs are grouped together and they are not necessarily within a clade. They can be monophyletic (a single clade), polyphyletic, or paraphyletic. Grouping information is useful to visualize specific groups or evolutionary paths. It can also serve as a factor or grouping variable for stratified analysis. Different *treedata* objects can be merged into one if the input trees have identical topologies. All the data stored in different tree objects can be mapped to the same tree topology to allow downstream integrative and comparative analysis.

Exporting a Phylogenetic Tree with Data to a Single File

Phylogenetic trees and associated data are often stored in separate files with inconsistent and incompatible formats. Using *treeio*, we are able to link the data and the tree into a single object. However, exporting a tree with associated data is still challenging. The major obstacle here is that there is no standard format designed for it. NeXML (Vos et al. 2012) is perhaps the most flexible format for storing tree and data, however, it is currently not widely supported. Most of the computer programs for analyses in this field rely extensively on Newick and NEXUS formats (Maddison et al. 1997). In our opinion, although the BEAST NEXUS format (Bouckaert et al. 2014) may not be the best solution, it is currently a sufficient approach for storing most heterogeneous-associated data. The best part of this format is that all the annotated elements are stored within square brackets, which are normally reserved for comments. This allows the file to be parsed as a standard NEXUS file by ignoring annotated elements. For more complicated scenarios, we propose using the *jtree* format as defined in *treeio* (see [Supplementary Material](#) online). The *jtree* format is based on JSON (JavaScript Object Notation). A *jtree* file can be easily parsed in any of the many modern programming languages that support JSON, whereas many programming languages still lack libraries or packages to parse BEAST NEXUS files.

The *treeio* package provides the *write.beast* and *write.jtree* to export *treedata* objects as BEAST NEXUS or *jtree* files, respectively. It also provides *as.phylo* to convert *treedata* object to a *phylo* object, which is widely supported in the R community and can be exported to Newick or NEXUS file using the *ape* package (Paradis and Schliep 2019). Combining the input and output utilities, *treeio* can be used as a phylogenetic tree file converter. For example, we can parse an output file of *CodeML* and export the tree with estimated d_N/d_S into the BEAST NEXUS format, which can be opened by *FigTree* and then d_N/d_S can be used to color the tree. Without *treeio*, the *CodeML* output cannot be opened by *FigTree* directly. File conversion is often necessary to effectively use multiple software packages in conjunction and *treeio* can serve as a connector between different software packages in an analysis pipeline.

Summary

To date, most software tools in the field of molecular evolution are isolated and often not fully compatible with each other's input and output files. These software tools are designed to do their own analyses and the outputs are often not readable in other software. No tools have been designed to unify the inference data from different analysis programs. An efficient incorporation of data from different inference methods can enhance the comparison and understanding of the study target, which may help discover new systematic patterns and generate new hypotheses. The *treeio* package provides a set of functions for parsing various types of phylogenetic data files. These phylogenetic data can be integrated to allow further exploration and comparison. Data mapped to a tree can be exported with the tree into a single file, which can enhance data sharing, reusability and analytic reproducibility.

As the use of phylogenetic trees to identify patterns in an evolutionary context grows, different disciplines are beginning to employ phylogenetic trees in their research. Mapping diverse data onto phylogenetic trees is increasingly central to a wide range of phylogenetic-related research. *Treeio* will help in integrating different data (time, geography, genotype, epidemiological information) and analysis results (selective pressure, evolutionary rates) to broaden our understanding of the study organisms. The *treeio* package is part of the *ggtree* package suite which contains the *tidytree*, *treeio*, and *ggtree* packages (Yu et al. 2017). The *tidytree* package converts tree objects to tidy data frames and vice versa, and provides a set of functions to manipulate trees with data using the tidy interface. The *ggtree* package is able to graphically represent tree-associated data on a phylogenetic tree. This full suite of packages makes it easy for anyone to interact and visualize trees with data.

Code Availability

treeio is freely available at <https://www.bioconductor.org/packages/treeio>. A complete reference of *treeio* is available in the online book, <https://yulab-smu.github.io/treedata-book/>, with many step-by-step instructions on phylogenetic tree and data input, output and manipulation as well as

visualization using *ggtree*. The *treeio* package is mainly described in the first three chapters.

Supplementary Material

Supplementary data are available at *Molecular Biology and Evolution* online.

Acknowledgments

We would like to thank Drs Scott Chamberlain, François Michonneau and Josef Uyeda (rOpenSci) and Prof. Martin Morgan (Bioconductor) for providing thorough code review, and the anonymous reviewers for comments on the article. This work was supported by research grants from National Key Plan for Scientific Research and Development of China (2016YFD0500302; 2017YFE0190800), Shenzhen Peacock Plan (KQTD201203), National Institutes of Health (HHSN272201400006C), NSFC-Guangdong Joint Foundation of China (U1601225), Open Project of Guangdong Provincial Key Laboratory of Proteomics (20191001A), and Startup fund from Southern Medical University (G618289088).

References

- Berger SA, Krompass D, Stamatakis A. 2011. Performance, accuracy, and web server for evolutionary placement of short sequence reads under maximum likelihood. *Syst Biol*. 60(3):291–302.
- Bouckaert R, Heled J, Kühnert D, Vaughan T, Wu C-H, Xie D, Suchard MA, Rambaut A, Drummond AJ. 2014. BEAST 2: a software platform for Bayesian evolutionary analysis. *PLoS Comput Biol*. 10(4):e1003537.
- Boussau B, Szollosi GJ, Duret L, Gouy M, Tannier E, Daubin V. 2013. Genome-scale coestimation of species and gene trees. *Genome Res*. 23(2):323–330.
- Höhna S, Landis MJ, Heath TA, Boussau B, Lartillot N, Moore BR, Huelsenbeck JP, Ronquist F. 2016. RevBayes: Bayesian phylogenetic inference using graphical models and an interactive model-specification language. *Syst Biol*. 65(4):726–736.
- Kumar S, Stecher G, Tamura K. 2016. MEGA7: molecular evolutionary genetics analysis version 7.0 for bigger datasets. *Mol Biol Evol*. 33(7):1870–1874.
- Maddison DR, Swofford DL, Maddison WP. 1997. Nexus: an extensible file format for systematic information. *Syst Biol*. 46(4):590–621.
- Matsen FA, Hoffman NG, Gallagher A, Stamatakis A. 2012. A format for phylogenetic placements. *PLoS One* 7(2):e31009.
- Matsen FA, Kodner RB, Armbrust EV. 2010. pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree. *BMC Bioinformatics* 11(1):538.
- Nguyen L-T, Schmidt HA, von Haeseler A, Minh BQ. 2015. IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Mol Biol Evol*. 32(1):268–274.
- Paradis E, Schliep K. 2019. ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics* 35(3):526–528.
- Pond SLK, Frost SDW, Muse SV. 2005. HyPhy: hypothesis testing using phylogenies. *Bioinformatics* 21(5):676–679.
- Rambaut A, Lam TT, Max Carvalho L, Pybus OG. 2016. Exploring the temporal structure of heterochronous sequences using TempEst (formerly Path-O-Gen). *Virus Evol*. 2(1):vew007.
- Ronquist F, Huelsenbeck JP. 2003. MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics* 19(12):1572–1574.
- Sanderson MJ. 2003. r8s: inferring absolute rates of molecular evolution and divergence times in the absence of a molecular clock. *Bioinformatics* 19(2):301–302.
- Stamatakis A. 2014. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* 30(9):1312–1313.
- Vos RA, Balhoff JP, Caravas JA, Holder MT, Lapp H, Maddison WP, Midford PE, Priyam A, Sukumaran J, Xia X, et al. 2012.

- NeXML: rich, extensible, and verifiable representation of comparative data and metadata. *Syst Biol.* 61(4):675–689.
- Wickham H, François R, Henry L, Müller K. 2019. dplyr: a grammar of data manipulation. R package version 0.8.3. <https://CRAN.R-project.org/package=dplyr>.
- Yang Z. 2007. PAML 4: phylogenetic analysis by maximum likelihood. *Mol Biol Evol.* 24(8):1586–1591.
- Yu G, Lam TT-Y, Zhu H, Guan Y. 2018. Two methods for mapping and visualizing associated data on phylogeny using Ggtree. *Mol Biol Evol.* 35(12):3041–3043.
- Yu G, Smith DK, Zhu H, Guan Y, Lam TT-Y. 2017. ggtree: an r package for visualization and annotation of phylogenetic trees with their covariates and other associated data. *Methods Ecol Evol.* 8(1):28–36.
- Zhang C, Rabiee M, Sayyari E, Mirarab S. 2018. ASTRAL-III: polynomial time species tree reconstruction from partially resolved gene trees. *BMC Bioinformatics* 19(S6):153.