

# Package ‘PhySortR’

May 7, 2026

**Type** Package

**Title** A Fast, Flexible Tool for Sorting Phylogenetic Trees

**Version** 1.0.9

**Date** 2025-02-07

**Maintainer** Timothy Stephens <timothy.stephens1@uqconnect.edu.au>

**Description** Screens and sorts phylogenetic trees in both traditional and extended Newick format. Allows for the fast and flexible screening (within a tree) of Exclusive clades that comprise only the target taxa and/or Non-Exclusive clades that includes a defined portion of non-target taxa.

**Depends** R (>= 3.0.0)

**Imports** phytools, ape, methods

**Repository** CRAN

**License** GPL (>= 3)

**RoxygenNote** 7.3.2

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Timothy Stephens [aut, cre, trl] (R port),  
Debashish Bhattacharya [aut],  
Mark Ragan [aut],  
Cheong Xin Chan [aut, cph] (Original Perl implementation)

**Date/Publication** 2025-02-13 00:20:02 UTC

## Contents

PhySortR-package . . . . .	2
convert.eNewick . . . . .	2
sortTrees . . . . .	3

<b>Index</b>	<b>6</b>
--------------	----------

---

PhySortR-package

*A Fast, Flexible Tool for Sorting Phylogenetic Trees*

---

### Description

**PhySortR** provides a quick and highly flexible function for the screening (within a tree) of Exclusive clades that comprise only the target taxa and/or Non-Exclusive clades that includes a defined portion of non-target taxa. Support is also provided for both traditional and extended Newick formatted phylogenetic trees.

A full list of functions can be displayed by `library(help = PhySortR)`.

### Details

Package: PhySortR  
Type: Package  
Version: 1.0.8  
Date: 2018-07-20  
License: GPL (>= 3)

### Author(s)

Timothy G. Stephens, Debashish Bhattacharya, Mark A. Ragan, Cheong Xin Chan

Maintainer: Timothy G. Stephens <timothy.stephens1@uqconnect.edu.au>

### References

Stephens TG, Bhattacharya D, Ragan MA, Chan CX. 2016. PhySortR: a fast, flexible tool for sorting phylogenetic trees in R. PeerJ, 4:e2038, DOI:10.7717/peerj.2038

---

convert.eNewick

*Converts Extended Newick Format to Traditional Newick Format*

---

### Description

Takes a phylogenetic tree in extended Newick format and converts it to traditional Newick format that can be directly manipulated by packages such as ape and phytools.

### Usage

```
convert.eNewick(eNewick)
```

**Arguments**

eNewick            phylogenetic tree in extended Newick format.

**Value**

phylogenetic tree in traditional Newick format.

**Examples**

```
### Converts the phylogenetic tree into traditional Newick format.
tree <- "((A:0.1,(B:0.3,C:0.2):0.2[60]):0.4[100]),(E:0.12,F:0.09):0.4[100]);"
new.tree <- convert.eNewick(tree)
new.tree
```

---

sortTrees

*Sorts Phylogenetic Trees using Taxa Identifiers*

---

**Description**

Reads phylogenetic trees from a directory and sorts them based on the presence of Exclusive and Non-Exclusive clades containing a set of given target leaves at a desired support value. Can interpret trees in both Newick and extended Newick format.

**Usage**

```
sortTrees(
  target.groups,
  min.support = 0,
  min.prop.target = 0.7,
  in.dir = ".",
  out.dir = "Sorted_Trees",
  mode = "l",
  clades.sorted = "E,NE",
  extension = ".tre",
  clade.exclusivity = 0.9
)
```

**Arguments**

target.groups    a set of one or more terms that represent the target leaves whose membership will be tested in each clade during sorting. Multiple terms are to be separated by a comma ("Taxon1,Taxon2"). This process is case sensitive and uses strict string-matching, so the taxa identifiers must be unique i.e. "plantae" and "Viridiplantae" might not be appropriate as the first is a subset of the second.

min.support     the minimum support (i.e. between 0-1 or 0-100) of a clade (Default = 0). Support values missing from phylogenetic trees are interpreted as zero. A vector of values can be provided if multiple support values (e.g., aLRT, UFboot) are present in the tree (i.e., "75.5/95").

<code>min.prop.target</code>	the minimum proportion (between 0.0-1.0) of target leaves to be present in a clade out of the total target leaves in the tree (Default = 0.7).
<code>in.dir</code>	directory containing the phylogenetic trees to be sorted (Default = current working directory).
<code>out.dir</code>	directory to be created within <code>in.dir</code> for the trees identified during sorting. If <code>out.dir</code> is omitted the default of <code>Sorted_Trees/</code> will be used.
<code>mode</code>	option to "m" (move), "c" (copy) or "l" (list) trees identified during sorting. In "l" mode (default) a list of the sorted trees is returned, in the "m" and "c" modes a list is returned and the identified trees are moved/copied to the <code>out.dir</code> .
<code>clades.sorted</code>	option to control if the function will sort for Exclusive ("E") and/or Non-Exclusive ("NE") clades. Specify both options by comma separation "E,NE" (Default). Exclusive clades are also sorted into a sub-group of All Exclusive trees.
<code>extension</code>	the file extension of the tree files to be analyzed (Default = ".tre").
<code>clade.exclusivity</code>	the minimum proportion ( $0.0 \leq x < 1.0$ ) of target leaves to interrupting leaves allowed in each non-exclusive clade (Default = 0.9).

### Value

Will always return a list containing the names of the trees identified during sorting, irrespective of the mode argument.

### Examples

```
### Load data ###
extdata <- system.file("extdata", package="PhySortR")
file.copy(dir(extdata, full.names = TRUE), ".")
dir.create("Algae_Trees/")
file.copy(dir(extdata, full.names = TRUE), "Algae_Trees/")

### Examples ###
# (1) Sorting using 3 target terms, all other parameters default.
sortTrees(target.groups = "Rhodophyta, Viridiplantae")

# The function will search in the users current working directory for files
# with the extension ".tre" and check them (using default min.support,
# min.prop.target and clade.exclusivity) for Exclusive, All Exclusive or
# Non-Exclusive clades. A list will be returned with the names of the trees
# identified during sorting.

# (2) Sorting with a target directory and an out directory specified.
sortTrees(target.groups = "Rhodophyta, Viridiplantae",
  in.dir= "Algae_Trees/",
  out.dir="Sorted_Trees_RVG/",
  mode = "c")

# The function will search in "Algae_Trees/" for files with the extension
```

```

# ".tre" and check them (using default min.support, min.prop.target,
# clade.exclusivity) for Exclusive, All Exclusive or Non-Exclusive clades.
# The function will both (a) return a list of the trees identified during
# sorting and (b) copy the files into their respective subdirectories of
# "Algae_Trees/Sorted_Trees_RVG/Exclusive/",
# "Algae_Trees/Sorted_Trees_RVG/Exclusive/All_Exclusive/" and
# "Algae_Trees/Sorted_Trees_RVG/Non_Exclusive/".

# (3) Sorting with in/out directories, min.prop.target and min.support specified.
sortTrees(target.groups = "Rhodophyta,Viridiplantae",
  min.prop.target = 0.8,
  min.support = 90,
  in.dir= "Algae_Trees/",
  out.dir="Sorted_Trees_RVG_95/",
  mode = "c",
  clades.sorted = "NE",
  clade.exclusivity = 0.95)

# The function will search in "Algae_Trees/" for files with the
# extension ".tre" and check them for only Non-Exclusive clades.
# A clade will only be defined if it has support >= 90 and contains at least
# 80% of the total target leaves in the tree. A Non-Exclusive clade must also
# be composed of >= 95% target taxa (i.e. < 5% non-target taxa).
# The function will (a) return a list of the trees identified during
# sorting and (b) copy the trees identified during sorting to the out
# directory "Algae_Trees/Sorted_Trees_RVG/Non_Exclusive/".

# (4) Sorting with multiple min.support values specified.
#sortTrees(target.groups = "Rhodophyta,Viridiplantae",
# min.prop.target = 0.8,
# min.support = c(75, 90),
# in.dir= "Algae_Trees/",
# out.dir="Sorted_Trees_RVG_75_95/",
# mode = "c",
# clades.sorted = "NE",
# clade.exclusivity = 0.95)

# The function will search in "Algae_Trees/" for files with the
# extension ".tre" and check them for only Non-Exclusive clades.
# A clade will only be defined if it has its first support >= 75
# and its second support >= 90 and contains at least 80% of the
# total target leaves in the tree. A Non-Exclusive clade must also
# be composed of >= 95% target taxa (i.e. < 5% non-target taxa).
# The function will (a) return a list of the trees identified during
# sorting and (b) copy the trees identified during sorting to the out
# directory "Algae_Trees/Sorted_Trees_RVG/Non_Exclusive/".

### Clean up ###
unlink("Algae_Trees", recursive=TRUE)
unlink("Sorted_Trees.log")
unlink(dir(".", "*.tre$"))

```

# Index

\* **package**

PhySortR-package, [2](#)

convert.eNewick, [2](#)

PhySortR-package, [2](#)

sortTrees, [3](#)