



Figure 1: logo

treeStartR

Phylogenetic trees, and particularly time-scaled phylogenetic trees, are increasingly estimated using complex models of evolution. Finding a starting tree with a computable likelihood to perform a Bayesian MCMC can be a challenge, particularly when estimation involves taxa, large datasets, many parameters, and missing data.

The purpose of this package is to allow users to efficiently add taxa to a given tree to generate a reasonable starting tree. While adding tips to a tree may be trivial if all tips have the same datatype (i.e., nucleotide sequence data), this can be more challenging for mixed datatype analyses. Additionally, taxa for which only stratigraphic data are available cannot have phylogenetic placements estimated, and must be placed via expert knowledge.

This package allows users to input a known phylogeny, and add tips to the tree. Tips can either be added via a pop-up prompt, in which users specify the exact node which the tip will subtend, or at random, or by providing a taxon set with which the tip should form a group.

Installation

Install the development version directly from Github

```
library(devtools)
install_github("wrightaprilm/treeStartR")
```

```
## Skipping install of 'treestartR' from a github remote, the SHA1 (5e619d08) has not changed since last
## Use `force = TRUE` to force installation
```

Dependencies

treeStartR depends on phytools.

```
library(phytools)
```

```
## Loading required package: ape
```

```
## Loading required package: maps
```

Usage Example

```
library(treestartR)
data(bears)
```

First, we need to load a list of the total set of taxa present in the tree. The “total set” refers to any taxa that will be included in your analysis. This can be either a CSV or a TSV file. A sample list, `tax_list`, has been provided as part of the bears data object, but you can also generate one using the function `dataf_parser`.

```
tax_list <- dataf_parser(path/to/tax_file)
```

Next, we find out which of the taxa from our total set are not represented on the tree already. This function takes as input a tree, with or without branch lengths, but without annotations (such as 95% HPDs). It also takes the total set of taxa generated by `dataf_parser`:

```
absent_list <- genera_strippr(tree, tax_list)
```

```
## [1] "Ailurarctos_lufengensis" "Indarctos_punjabiensis"
## [3] "Kretzoiarctos_beatrix"  "Parictis_montanus"
## [5] "Ursavus_brevirhinus"    "Ursavus_primaevus"
## [7] "Ursus_abstrusus"
```

Adding tips with congeners

Finally, we add the tips that are not present to the tree. If there are other representatives of the same genera as an absent taxon (for example, adding an additional “Ursus” species to the example tree), those taxa will be used to place the tip. If there are multiple species of the genera, the new tip will subtend the most recent common ancestor of the tips already on the tree. If there is only one representative, the tip will subtend the parent node of that taxon.

```
new_tree <- present_tippr(tree, absent_list)
```

```
## [1] "Adding tips with congeners on tree:"
## [1] "Indarctos_punjabiensis"
```

```
## Warning in bind.tree(tree, tip, where = where, position = pp): one tree has
## no branch lengths, they have been ignored
```

```
## [1] "Adding tips with congeners on tree:"
## [1] "Ursus_abstrusus"
```

Adding tips manually

We can also add the tips that have no congeners. This function will ask for input. A pop-up will be produced, showing node labels. When the program asks for input, you will tell it what node you would like the tip to subtend:

Adding tips at random

Or, if there are no congeners, you may choose to add tips at random:

```
new_tree <- rand_absent_tippr(tree, absent_list)
```

```
## [1] "Ailurarctos"
## [1] "Indarctos"
## [1] "Kretzoiarctos"
## [1] "Parictis"
## [1] "Ursavus"
## [1] "Ursavus"
## [1] "Ursus"
```

Adding tips via CSV

Lastly, you may have a TSV file that specifies the tips to be added, and a taxon set. `treeStartR` will locate the MRCA of the taxon set, and add the tips subtending that node.

```
new_tree <- text_placr(tree, mrca_df)
```

```
## [1] "Placing taxon via provided input: Kretzoiarctos_beatrix"
## [1] "Placing tip at node 25"

## Warning in bind.tree(tree, tip, where = where, position = pp): one tree has
## no branch lengths, they have been ignored

## [1] "Placing taxon via provided input: Ursus_abstrusus"
## [1] "Placing tip at node 30"
```

Outputting results

The final tree can be output using standard functions in `ape`, such as `write.nexus()`: