Design decisions for the phylo4 class

Ben Bolker

December 11, 2007

This document describes the design decisions associated with the new "phylo4" class, which is intended to provide some kind of unifying standard for phylogenetic data in R. It is closely modeled on the the phylo class in ape, which is the dominant data structure at the moment.

Like phylo, the main components of the phylo4 class are:

```
edge an N \times 2 matrix of integers, where the first column ...
```

edge.length numeric list of edge lengths (length N or empty)

Nnode integer, number of nodes

tip.label character vector of tip labels (required)

node.label character vector of node labels (maybe empty)

root.edge integer defining root edge (maybe NA)

Print method: add information about (ultrametric, scaled, polytomies (zero-length or structural))?

1 Hacks/backward compatibility

Hilmar Lapp very kindly showed a way to hack the \$ operator so that it would provide backward compatibility with code that is extracting internal elements of a phylo4. The basic recipe is:

```
> setMethod("$", "phylo4", function(x, name) {
+ attr(x, name)
+ })
```

but this has to be hacked slightly to intercept calls to elements that might be missing. For example, ape detects whether log-likelihood, root edges, node labels, etc. are missing by testing whether they are NULL, whereas missing items are represented in phylo4 by zero-length vectors in the slots (or NA for the root edge) — so we need code like

```
> if (!hasNodeLabels(x)) NULL else x@node.label
```

to handle these cases.

> library(ape)

2 To do/problems

- Conflict with nTips if ape is loaded first: ask EP to get rid of this (obsolete?) function? (Ntips is the real ape function for getting the number of tips)
- basic tree manipulation: tip-dropping, na.omit, etc. especially for multi-tree and tree-with-data cases
- tree-manipulation code: tree traversal (store current position as an attribute), pruning, etc.