

The `newicktree` package for phylogenetic trees

George Savva

March 1, 2004

This document contains the documentation for the March 2004 release of the package.

1 Motivation

Until now, including a figure of a phylogenetic tree in a \LaTeX document has meant loading the treefile into some other program such as TreeView, Tree-Explorer or drawtree, printing the tree to a file, extracting an encapsulated postscript image, and then importing the image into the document using an `\includegraphics` command.

If the tree needed to be changed for any reason, even one as trivial as a spelling mistake or to change formatting or scaling, the entire process would have to be repeated, for one or potentially many of the trees in a document. If the document contains a lot of trees, this would be very boring, and it would become difficult to maintain a consistent look to the trees.

The package `newicktree` eliminates these problems by providing direct rendering of Newick format trees via the `pstricks` package. This approach also has the advantage of providing high quality \TeX typesetting for tree labels, making them consistent with the rest of the document. The advantage of using `newick-tree` over `pstricks` directly is that it is much easier to describe a phylogenetic tree in Newick format than in `pstricks` code, and the options and defaults have been set up with representations of phylogenies in mind. The disadvantage is that much of the power of `pstricks` is lost, so if you need a unusual complicated diagram or you would like a lot of control you should invest some time in learning a lower level package.

2 Basic Usage

The format for including a phylogenetic tree in a \LaTeX document using `newick-tree` is

```
\usepackage{...,newicktree,...}  
  
...  
  
\begin{newicktree}  
  \option1  
  \option2
```

```

..
\optionn
\drawtree{<Newick format tree here>}
\end{newicktree}

```

The `\drawtree` command is responsible for the actual drawing of the tree, and must be placed inside a `newicktree` environment. As many `drawtree` commands as are required can be placed within a single `newicktree` environment. The option commands control the way any trees in the environment subsequent to the option command are drawn. Examples of options are `\downtree` and `\showbranchlengths` - these will be discussed in more detail later.

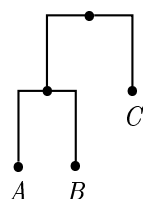
Let us look now at some simple examples of the use of this package. The example below uses `newicktree` to include a very simple tree in a document. Including

```

\begin{newicktree}
\drawtree{((A,B),C);}
\end{newicktree}

```

in your `TEX` file causes the following tree to be rendered.



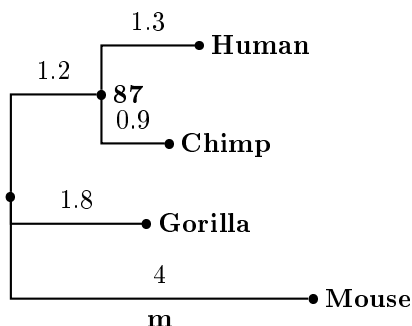
Trees may be specified with branch lengths, and extra edge labels or labels for the internal nodes may be provided. For example, the tree below has branch lengths included, an edge label (the 'm') on the mouse branch, and a node label (the '87' at the human / monkey node).

```

\begin{newicktree}
\righttree \nodelformat{\bfseries}
\drawtree{((Human:1.3,Chimp:0.9)87:1.2,Gorilla:1.8,Mouse:4[m]);}
\end{newicktree}

```

produces



Notice also the use of the `\righttree` command, which causes the tree to be drawn left-to-right (default is `\downtree`), and `\nodelformat`, which controls the way taxon names are drawn.

3 Command List

In the previous section two basic uses of the package were illustrated. This section contains a full survey of `newicktree`'s commands. Each option command changes one or more internal variables, which are used when the trees are drawn. Options can be overridden within a `newicktree` environment. The scope of an option is restricted to the particular `newicktree` environment within which it is specified, unless its first letter is capitalised, in which case it is global to the rest of the document. Commands can only be issued inside a `newicktree` environment; this helps to prevent clashes with other packages.

3.1 Options

`\uptree` `\downtree` `\lefttree` `\righttree`

These commands specify the direction of growth of the tree. The default is `\downtree`.

`\showbranchlengths` `\nobranchlengths`

`\showbranchlengths` causes branch lengths to be drawn on the tree if they are supplied. `\nobranchlengths` inhibits this, and causes the nodes to be drawn slightly closer together. Length labels will be drawn above or to the right of the branch to which they correspond. The default setting is `\showbranchlengths`.

`\usebranchlengths` `\unitbranchlengths` `\contemporarytips[.]`

These options are used to determine whether or not the branch lengths provided in the tree are used for drawing. If so, then the branches are drawn to the scale determined by the `\setunitlength` command. There are two options that cause the true lengths to be ignored, these are the `\unitbranchlengths` command, which sets every branch length to the value specified by the most recent `\setunitlength` command, and the `\contemporarytips` command, which ensures that all of the terminal nodes on your tree line up (see also the `\highlowlabels` and `\settreesep` commands). These options are useful if the actual lengths within a tree are vastly different from one another, and a display of a topology is all that is required. This does not affect the behaviour of the `\showbranchlengths` command, so the true lengths can still be displayed.

The `\contemporarytips` command has an optional argument that specifies the percentage of the remaining tree height is used by successive levels. This must be an integer between 1 and 99, and if this is not specified a value of 41 will be used. See example five in the gallery for an illustration.

The default option is `\usebranchlengths`.

`\highlowlabels` `\nodeseparation{<dimen>}`

Sometimes you will want more control over the spacing of nodes in a tree, particularly if you require a vertical tree with contemporary tips. The `\nodeseparation` command gives you that control, allowing you to override the default spacing between successive tree nodes. The spacing is taken in addition to the size of the node labels, so a setting of zero will cause the node labels to run together continuously. You may need to use a negative value to get your tree to fit in a narrow horizontal space properly, but this will cause the node labels to overlap. For this reason the `\highlowlabels` command is provided, which alternately draws node labels 2ex lower then higher than their neighbour (see example 2 in the tree gallery).

`\showbootstrapvalues` `\nobootstrapvalues`

`\showbootstrapvalues` causes the bootstrap values (or extra branch labels) associated with each branch to be drawn, and `\nobootstrapvalues` inhibits this. Bootstrap labels will be drawn below or to the left of the branch to which they correspond. The default setting is `\showbootstrapvalues`.

`\nodemarkers` `\nonodemarkers`

These options to switch on or off the large round blobs at the internal and leaf nodes. The default is on.

`\setunitlength{<dimen>}`

This option changes the scale of the tree. The scale is the length of a branch of size 1 in the units used to describe the tree. `<dimen>` can be any valid TeX dimension, such as 1cm, 3.5in, 10pt, etc. Default is 1cm.

`\odelabelformat{<format>}` `\lengthlabelformat{<format>}`
`\bootlabelformat{<format>}`

These commands change the way the text in the tree is drawn. For example, if `\odelabelformat{\itshape}` is used, the taxon names at the terminal nodes will be italicised. The format commands can take more than one parameter, indeed anything in the argument is effectively inserted in front of each node label before it is typeset. Note that, subsequent calls to `\odelabelformat`, say, will completely overwrite previous definitions, so to get bold sans-serif labels, `\odelabelformat{\bfseries \sffamily}` should be used, overwriting our previous italic font. The defaults are italic node labels, emboldened bootstrap labels and plain length labels. To affect a particular node label L^AT_EX markup can be included in the Newick tree itself.

`\angledbranches` `\curvedbranches` `\straightbranches`

These control the style of the edges used to connect the nodes. They do not change the positions of the nodes themselves. Currently the only ‘safe’ option is angled branches, and this is the default. Using a curved or single

straight line branch can look attractive and produce the desired result, but may cause alignment problems if you try to use branch labels, and in some cases, particularly with trifurcations, the branches might cross. The curved and straight branches are at their safest when used with the `\unitbranchlengths` option.

3.2 Other commands

These commands are not options and are responsible for the actual drawing of tree diagrams. As a consequence they do not have capitalised versions.

`\lb \rb`

Due to the way `newicktree` parses the Newick format, you cannot type (or) normally within a `newicktree` environment. The commands `\lb` and `\rb` are included for this reason. `\lb` produces a left bracket, `\rb` produces a right bracket.

`\scalebar[<length>]`

The `\scalebar` command causes a scalebar to be drawn at the current point. The optional `<length>` specifies the length (in tree units) of the bar. If none is specified, `newicktree` draw a bar of length 1 in the tree units.

`\drawtree[<root node label>]{<newick format tree>}`

The `\drawtree` command causes a tree to be drawn at the current point, using all of the options set prior to the command. Any text in the optional argument is interpreted as a root node label, and is added to the root node using whatever format is currently specified by the `\nodelabelformat` command. Only one tree is allowed per `\drawtree` command, and that tree must be terminated by a semi-colon. For more information about the tree specification, see the Troubleshooting and Known Issues section below.

3.3 Commands not defined by `newicktree`

`\scalebox{<h-scale>}[<v-scale>]{ ... }`
`\resizebox*{<h-dim>}{<v-dim>}{ ... }`
`\rotatebox{<angle>}{ ... }`

These commands belong to the \LaTeX `graphics` (and `graphicx`) packages. They allow the resizing and rotating of any \LaTeX material. The `\scalebox` command scales the contents horizontally by `h-scale` and vertically by `v-scale`. If the optional `v-scale` is omitted, the same scale factor is used for both dimensions and so aspect ratio is maintained.

The `\resizebox` command resizes the contents to the specified size. If either `v-dim` or `h-dim` is replaced by and exclamation mark (!), the same scale factor is applied to both dimensions so aspect ratio is maintained.

Rotating is accomplished using the `\rotatebox` command. This rotates material clockwise by the angle specified. For more detail about any of these and the other \LaTeX graphics commands, see the graphics documentation or consult a book such as *The \LaTeX Graphics Companion*.

`\psset{...}`

`newicktree` uses the tree drawing routines of `PSTricks` to draw trees, consequently `\psset` commands can be issued directly to effect changes to the way trees are drawn. Some examples of this are `\psset{linewidth=0.5mm}` and `\psset{arrows=->}`. Such commands are always local to the environment in which they are specified. See the `PSTricks` documentation for more details.

4 How does the package work?

Some of the details of how `newicktree` works are given below as they have important consequences, or they will help you to get the best from the package. For more information, contact me at `george.savva@bbsrc.ac.uk`

4.1 The `PSTricks` interface

`newicktree` uses the `pst-tree` and `pst-node` subpackages of `PSTricks` to render trees. More specifically, trees are drawn with the `\pstree` command making heavy use `\skiplevel`, which enables branches with differing lengths to descend from the same node.

An up to date version of `PSTricks` must be used in conjunction with `newicktree`. This is not usually a problem since `PSTricks` is well used and is included in most reasonably large \LaTeX distributions (it is, for example, included in the default set up for the \LaTeX and \TeX distributions), and it is easy to obtain from CTAN if you do not have it.

`PSTricks` provides an interface to the PostScript language, so to view trees (or indeed any PostScript graphic) correctly, you must convert your dvi files to PostScript, using a program such as `dvips`. Consequently, you cannot use `pdfLaTeX` to compile your documents. If you require a pdf document, use `dvips -ppdf` (causing `dvips` to use outline fonts) and then `ps2pdf` (part of `GhostView`) to generate the pdf. This approach was used to generate the pdf version of this document.

4.2 Getting `newicktree` and making it work

All of the code for the `newicktree` package is found in the `newicktree.sty` file, which can be downloaded from its website¹. The file needs to be in some place that your \LaTeX installation will find, for example in the local \TeX tree or in the current working directory. See the instructions for installing packages that came with your distribution, or contact your computing department for more information.

¹<http://jic-bioinfo.bbsrc.ac.uk/bioinformatics-research/software/newicktree/>

In the preamble of your document you need to specify that the package `newicktree` is to be used (within a `\usepackage` command). This will automatically load `PSTricks`, and specifically `pst-tree` and `pst-node`, which are used in the package.

4.3 Using `newicktree` with plain \TeX .

The majority of the `newicktree` code uses only plain \TeX so it is possible, with some manipulation, to make `newicktree` work outside of \LaTeX . In the future a `newicktree.tex` file may be provided for this purpose. Contact me if you would be interested in this.

5 Tree gallery

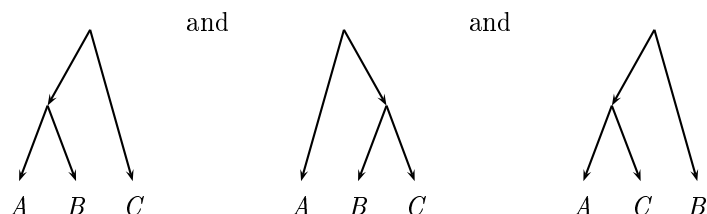
Below are some examples of what is possible using the `newicktree` package.

Example 1

The three different rooted trees relating three taxa are:

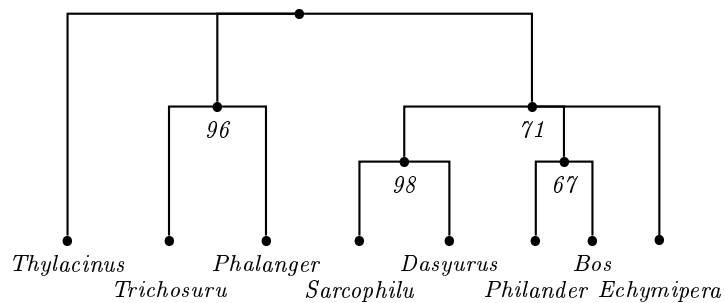
```
\begin{center}\begin{newicktree}
\straightbranches\nobranlengths\nonodemarkers\contemporarytips[50]
\psset{arrows=->}\setunitlength{2cm}
\drawtree{((A,B),C);} \quad and \quad \drawtree{(A,(B,C));}
\quad and \quad \drawtree{((A,C),B);}
\end{newicktree}\end{center}
```

The three different rooted trees relating three taxa are:



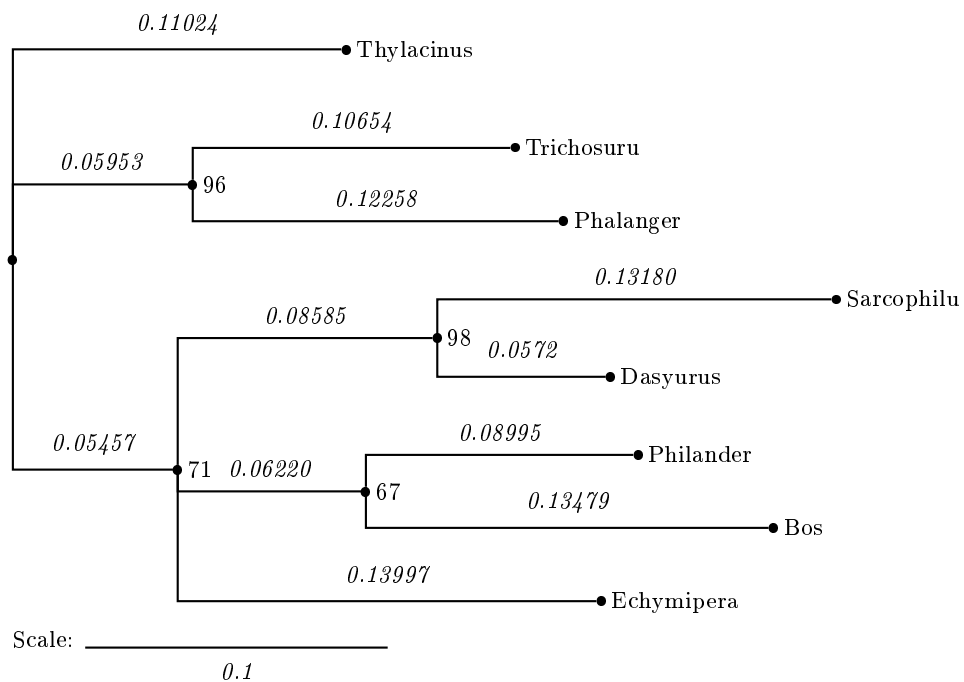
Example 2

```
\begin{newicktree}
\small
\downmtree \nobootstrapvalues \nobranlengths
\contemporarytips \highlowlabels \nodeseparation{-.2cm}
\setunitlength{3cm} % Sets the length of the whole tree in this case
\centerline{\drawtree{(Thylacinus:0.11024,(Trichosuru:0.10654,Phalanger:0.12258)96:0.05953,
((Sarcophilu:0.13180,Dasyurus:0.00572)98:0.08585,(Philander:0.08995,Bos:0.13479)67:0.04220,
Echymipera:0.13997)71:0.05457);}}
\end{newicktree}
```

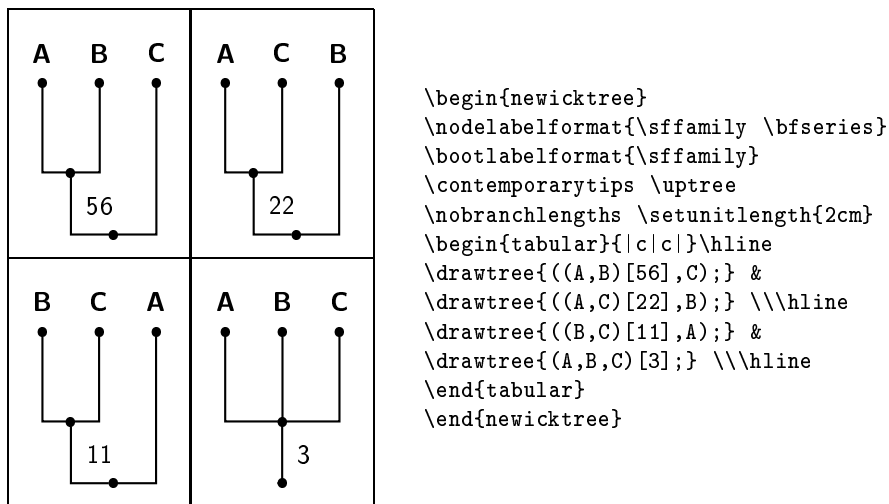


Example 3

```
\begin{newicktree}
\small
\setunitlength{40cm} \righttree \usebranchlengths
\lengthlabelformat{\it} \odelabelformat{}
\drawtree{(Thylacinus:0.11024,(Trichosuru:0.10654,Phalanger:0.12258)96:0.05953,%
((Sarcophilu:0.13180,Dasyurus:0.0572)98:0.08585,(Philander:0.08995,Bos:0.13479)%
67:0.06220,Echymipera:0.13997)71:0.05457);}
\par Scale: \scalebar[0.1]
\end{newicktree}
```



Example 4

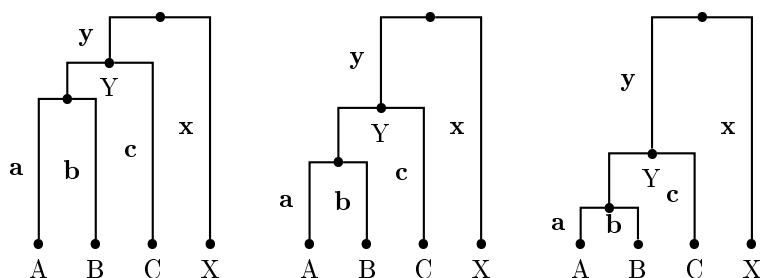


Example 5

```

\begin{newicktree}
\setunitlength{3cm}\nobranlengths\nodelabelformat{}
\contemporarytips[20]\drawtree{((A[a],B[b]),C[c])Y[y],X[x];}
\contemporarytips[40]\drawtree{((A[a],B[b]),C[c])Y[y],X[x];}
\contemporarytips[60]\drawtree{((A[a],B[b]),C[c])Y[y],X[x];}
\end{newicktree}

```

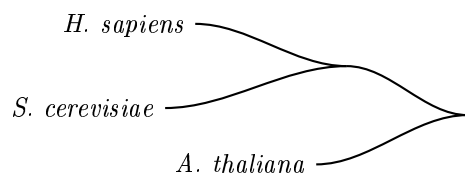


Example 6

```

\begin{newicktree}
\curvedbranches\nobranlengths\nonodemarkers
\lefttree\setunitlength{.2cm}
\drawtree{(H. sapiens:10,S. cerevisiae:12):8,A. thaliana:10;}
\end{newicktree}

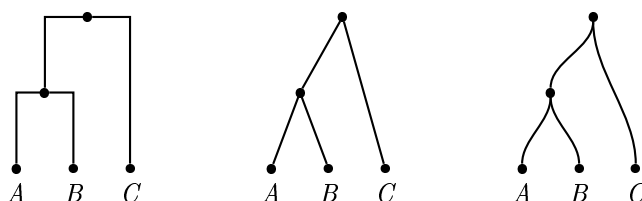
```



Example 7

```
\begin{center}
\begin{newicktree}
\nobranlengths
\begin{tabular}{c c c}
\verb=\angledbranches=&\verb=\straightbranches=&\verb=\curvedbranches=\\
\angledbranches\drawtree{((A:1,B:1):1,C:2);}&
\straightbranches\drawtree{((A:1,B:1):1,C:2);}&
\curvedbranches\drawtree{((A:1,B:1):1,C:2);}
\end{tabular}
\end{newicktree}
\end{center}
```

`\angledbranches` `\straightbranches` `\curvedbranches`



6 Troubleshooting

Most problems with the use of `newicktree` arise from trying to use badly specified trees. Some points to remember:

- Trees must be properly specified. In particular the brackets need to match. If you are using a text editor with bracket matching then it will be easier to see if something is wrong.
- Trees must be terminated by a semi-colon.
- Trees may not contain newlines. If they do, maybe due to the way the treefile is generated, then ensure every newline is suppressed by a comment character (see bugs section below).
- You will not see your tree correctly if you do not convert your document to PostScript. Most dvi viewers do not deal with PostScript entirely correctly.

7 Future developments / wishlist / changelog

7.1 March 2004 Version

The following changes were made for the March 2004 release of the package

- Trees without distances are now allowed.
- Internal node labels are now drawn at the nodes.
- The optional argument for the contemporary nodes option was introduced.

7.2 The future

A future release of the package is planned, if the idea seems sufficiently popular with its users. The options and code format will be completely compatible, but the tree drawing system will be reconsidered in the light of the experience gained from writing this first version.

Also, particular features that are missing from `newicktree` will be added.

- Inputting trees directly from files will be supported.
- An option to drawn radial trees will be added.
- Specifying the total size of the tree instead of the scale in all modes.
- Improved flexibility in the labelling of edges and nodes.
- Improved labelling of edges in straight / curved mode.

7.3 Known Issues

- Including spaces in the treefile (except in the taxon names) can cause problems. This applies also to new lines. Avoid any white space or linebreaks in the treefile. If you must have linebreaks then terminate each line with the comment character `%` (see example 2). This is general good practice anyway.
- `TEX` has a maximum size for dimensions that cannot be exceeded. This size is about twenty feet, which isn't usually a problem when creating `TEX` documents, but becomes a problem in `newicktree` when you are trying to describe a tree with very short branches, as the `\setunitlength` command might fail. Two ways to avoid to this problem are
 - Use the `\contemporarytips` or `\unitbranchlengths` options to produce a topology
 - Make the original distances in your tree larger, by replacing every occurrence of 0.00 with 0.0 or something similar in your Newick tree. Your text editor might make this job easy if it has a search-and-replace function.

This issue is at the top of the priority list for the next release of `newicktree`

8 Acknowledgements / IP / Contact

This work is supported by the John Innes Foundation. `newicktree` is released under the L^AT_EX Project Public License²

If you have any problems, bug reports, suggestions or other useful comments, contact me at `george.savva@bbsrc.ac.uk`.

Also email me if you would like to kept up to date with any future verions of `newicktree`.

²<http://www.latex-project.org/lppl.html>