PareTree 1.0.2

Emma Hodcroft
Andrew Leigh Brown Group
Institute of Evolutionary Biology
University of Edinburgh

This command-line Java program allows users to ‘pare’ down their tree by either removing unwanted leaves (tip-nodes), removing bootstrap information from the tree, or removing branch lengths from the tree – or any combination. These functions can be accomplished in languages like R or Perl, but Java allows very large trees to be pared down quickly, efficiently, and easily!

The program takes Nexus/Newick-style phylogenetic tree code as input that may or may not include bootstraps, and outputs Newick-style trees with the requested nodes and/or bootstrap values removed.

Much like its sister-program, TreeCollapseCL, this program accepts two types of file - to see details on these types, please see ‘`-t`’ below.

PLEASE NOTE: I am not responsible for any incorrect behaviour of this program. I do not guarantee that it will behave correctly or as you predict because I cannot test it in every conceivable situation.

Reminder:
Deleting nodes from phylogenies is risky business! Deleting nodes is unlikely to give you the same phylogeny as re-running the tree with the corresponding sequences removed. You could easily misrepresent your data to yourself and to others by deleting nodes to make a ‘prettier’ picture. Thus, I implore the user to run this program carefully and **I do not condone** it being used to intentionally misrepresent data!

Here are a few (safer) situations where this program might be useful:

- Deleting identical sequences

  Some phylogenetic programs do not warn of identical sequences at the beginning of a run, leading to leaf-pairs with branch-lengths of 0. Some programs have trouble with or will not accept branch lengths of 0. If the tree is large and the run takes many hours or days, deleting the identical sequences and re-running the tree (or trees) can be a very unappealing task. Because both leaves have a branch length of 0 from the previous node, deleting one will not affect the legitimacy of your tree, and can save a lot of time. Also, the program can be put into a batch file or script so that the same sequences are deleted from many tree files quickly and easily.
- ‘Zooming In’ on Trees / Simplifying Outgroups

  For display purposes, PareTree could be used to delete all but a cluster of sequences from a tree, allowing a better, ‘zoomed-in’ image of a section of the tree the tree for easier examination. Similarly, if multiple, monophyletic outgroup sequences have been used, but are affecting the display of the tree, all but one could be deleted (and then this one renamed to reflect that it was originally a group of sequences).

It is worth mentioning that the total root-to-tip distance is kept the same for every leaf after other leaves are deleted. (Non-deleted nodes will have the same total distance to the root as they did before. See the page 4 for more details.)

Updates since version 1.0.1: (Previous updates on last page)

- '-topo' can now be used to specify that output files should be written without branch lengths (topology only). Can be combined with -nbs and -del/-keep or used alone
- Now compatible with Windows Shell auto-glob when passing * to specify file endings
- '-t' is now optional - if not included the value defaults to 'O'
- '-rax' is now depreciated (but can be included without affecting the run). The program now automatically detects and handles a larger number of minor variations in format that can occur at the end of Newick files

Parameter Notes:

The parameters '-d' *or* '-f' MUST be supplied, as well as '-del' *or* '-keep' and/or '-nbs' and/or '-topo'. The parameters '-v' and '-t' are optional. The parameter '-rax' is now not needed and is depreciated.

```
java -jar PareTree.jar  -t
                        -d dir or -f file
                        [-del] leavesToDelete
                        [-keep] leavesToKeep
                        [-nbs]
                        [-topo]
                        [-v]
                        [-rax]
```

If the program is run without any parameters, a list of parameters with descriptions is displayed.

Some tips on running multiple files can be found at the end of this document, as well as citation information.

Parameters:

`-t` (optional)

Use '-t' to specify the file type that will be read in.

-t O (Default) Use 'O' (capital o) to specify that the file is the usual Newick/Nexus-type file, with bootstrap values preceding colons. Unless the file was exported as Nexus with annotation in FigTree, use this option (or don't include the parameter at all)!
Example: "`((B:0.04,C:0.03)0.83:0.01);`" Where 0.83 is the bootstrap value.

-t F Use 'F' to specify that the file is a Nexus-type file that's been exported from FigTree with annotations. These have bootstrap values within square brackets ('[]').
Example: "`((B:0.04,C:0.03)[&bs=0.83]:0.01);`" Where 0.83 is the bootstrap value.

`-d dir` *or* `-f file`

These specify the file (use '-f') or directory (use '-d') of files to be read in.
Follow '-f' with the file name.
Example: `java -jar PareTree.jar -t O -del Mouse_1990 -f Sequences.newick`
Follow '-d' with the directory containing the files to be read in. %CD% can be used as well. If there may be spaces in folders or filenames in the path, use double quotes ("") to enclose the path. It's a good idea to use these unless you're certain there are no spaces.
Example: `java -jar PareTree.jar -t O -del Mouse_1990 -d C:\Users\Bob\Sequences`
(Continued…)

Example: `java -jar PareTree.jar -t O -del Mouse_1990 -d “%CD%”`
You can also specify the ending of the files to be read by using '*.' followed by the ending. Be aware that this will only work on endings - putting something before the * will not work.
Example: `java -jar PareTree.jar -t O -del Mouse_1990 -d C:\Users\Sequences\*.newick`
Example: `java -jar PareTree.jar -t O -del Mouse_1990 -d “%CD%\*.nexus”`

`-del leavesToDel` *or* `-keep leavesToKeep`

<u>Use one or the other, not both!</u> This parameter specifies the sequences that should be either deleted (using '`-del`') or kept (using '`-keep`'). The user can specify either a single sequence name directly in the run command, or specify a file that contains a list of sequences, one per line, no punctuation. When using '`-del`', all sequences **<u>in</u>** the file will be deleted. When using '`-keep`', all sequences **<u>not</u>** in the file will be deleted. Any sequences in the file that are not in the tree will be ignored in both cases.
If you submit an empty file (or a file where none of the sequences are in the tree) using '`-keep`' the program will terminate with an error. If you submit just one sequence using '`-keep`' the program will execute, but many other programs will not accept the resulting single-node ‘tree.’
The resulting tree will be written to a file with '`_pared`' added to the file name.
Example: `java -jar PareTree.jar -keep goodSeqs.txt -t O -f Sequences.newick`
Will delete all sequences **<u>not</u>** in the file '`goodSeqs.txt`' and output the resulting tree in '`Sequences_pared.newick`'
Example: `java -jar PareTree.jar -del badSeqs.txt -t O -f Sequences.newick`
Will delete all sequences **<u>in</u>** the file '`badSeqs.txt`' and output the resulting tree in '`Sequences_pared.newick`'
Example: `java -jar PareTree.jar -del Mouse_1990 -t O -f Sequences.newick`
Will delete the sequence named '`Mouse_1990`' and output the resulting tree in '`Sequences_pared.newick`'

`-nbs`

Use this to specify that any output tree files should NOT contain bootstrap values. If '`-del`' or '`-keep`' has been used, the resulting tree will automatically be returned without bootstrap values. If they have not, the original tree will be returned without bootstraps, with '`_nbs`' added to the file name.
This can be useful if the user plans to use some functions in the ‘`R`’ packages ‘`ape`’ or ‘`MCMCglmm`,’ as bootstrap information can cause some functions to work incorrectly.
Example: `java -jar PareTree.jar -nbs -t O -f Sequences.newick`
Will output a file called '`Sequences_nbs.newick`' which will not contain bootstrap values
Example: `java -jar PareTree.jar -nbs -del badSeqs.txt -t O -f Sequences.newick`
Will delete all sequences **<u>in</u>** the file '`badSeqs.txt`' and output the resulting tree in '`Sequences_pared.newick`' – without bootstrap values

`-topo`

Use this to specify that any output tree files should NOT contain branch lengths (that it should be a topology only). If '`-del`' or '`-keep`' has been used, the resulting tree will automatically be returned without branch lengths. If they have not, the original tree will be returned without branch lengths, with '`_topo`' added to the file name. If ‘`-nbs`’ has been used, it will be returned without branch lengths and without bootstrap values, with ‘`_nbs`’ added to the name.
Example: `java -jar PareTree.jar -topo -t O -f Sequences.newick`
Will output a file called '`Sequences_topo.newick`' which will not contain branch lengths
(Continued…)

Example: `java -jar PareTree.jar -topo -del badSeqs.txt -t O -f Sequences.newick`
Will delete all sequences **in** the file '`badSeqs.txt`' and output the resulting tree in '`Sequences_pared.newick`' – without branch lengths

`-v` (optional)

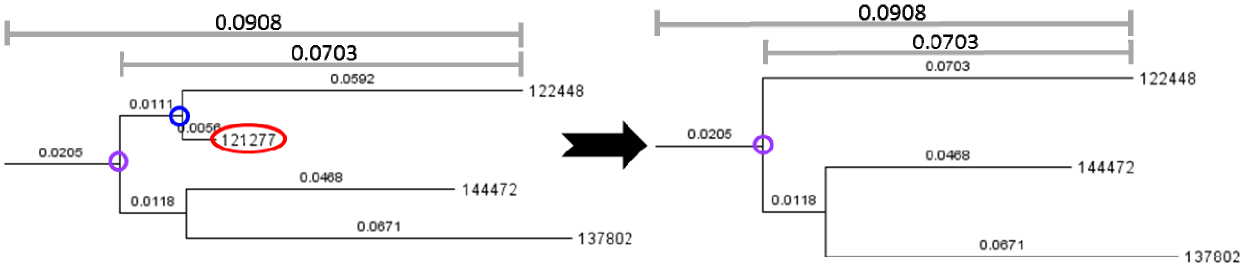
Use this to turn on very crude 'debug' which will basically output intermediate flags and steps to the console. It's probably not very useful, and may slow down runtime and may possibly even crash the run if turned on for very large runs with large files. Again, it's probably not very useful.

`-rax` (depreciated)

This used to be required to specify the specific format of a Newick file. The program now detects this automatically. For backwards compatibility, it can be included, but it will not affect the run.

A Note on Branch Length:

Leaves remaining after deletion retain the same total distance from the root of the tree and from other internal, ancestral nodes unaffected by the deletion. This is done by adding the branch lengths of deleted internal nodes (deleted because a leaf was deleted and there is no longer need for an internal bifurcation node) are added to the branch lengths of remaining nodes. For example:



Here, leaf 121277 (red) has been deleted from the tree. This lead to an internal node (blue) being deleted as well. However, the total distance from the root to all the other leaves remains unchanged. The distance from 121277's 'sister' node, 122448, to the next unaffected internal ancestral node (purple) also remains unchanged.

Tips on running multiple files:

If the user needs to run multiple files as input (by specifying a directory ('`-d`')), ensure they all have different names. If they have been generated by a batch run or are otherwise likely to have similar names, this program is designed to handle files that are numbered with the number in between decimal points preceding the file ending, as shown:

```
SeqSet_run.1.newick
SeqSet_run.2.newick
```

This then allows you specify a directory where all files (or all files with a specific ending) will have the same sequences deleted, as specified by '`-del`' or '`-keep`', or all have their bootstrap values removed. Output files are modified so that the file name change is put *before* the number in the file (ex: 'SeqSet_run_**pared**.1.newick'; 'SeqSet_run_**nbs**.2.newick').

Example: `java -jar PareTree.jar -t O -del Mouse_1990 -d C:\Users\Sequences\*.newick`
Example: `java -jar PareTree.jar -t O -nbs -keep goodSeq.txt -d "%CD%\*.nexus"`
Example: `java -jar PareTree.jar -t O -nbs -topo -d C:\Users\Sequences\*.newick`

Citation & Feedback:

If you publish or present work that has been processed using this program, please cite Emma Hodcroft and the website where this program can be downloaded (http://emmahodcroft.com/PareTree.html).

If you have questions about using the program, or would like to provide feedback or suggestions, please use the Feedback page on my website: http://emmahodcroft.com/feedback.html.

Updates from Previous Versions:

Updates since version 1.0:

- Corrected file/directory reading for Unix/Linux/Mac users
- Corrected error that assumed all files had a path before the file name