

MPBoot version 1.1.0 (August 2017)

MPBoot: Fast phylogenetic maximum parsimony tree inference and bootstrap approximation

User Manual

Project members:

Diep Thi Hoang - diep.thi.hoang(at)gmail.com; diepht(at)vnu.edu.vn

Le Sy Vinh - vinhls(at)vnu.edu.vn

Tomas Flouri - tomas.flouri(at)h-its.org

Alexandros Stamatakis - alexandros.stamatakis@h-its.org

Arndt von Haeseler - arndt.von.haeseler(at)mfpl.ac.at

Bui Quang Minh - minh.bui(at)mfpl.ac.at

1 Introduction

MPBoot is an efficient program for parsimony analysis. The program can reconstruct maximum parsimony trees for large DNA and protein sequence alignments. More importantly, it implements the method MPBoot for approximating maximum parsimony bootstrap. The method was inspired by the ultrafast bootstrap approximation for maximum likelihood (UFBoot; Minh et al. 2013) and other local optimization techniques for tree search. MPBoot is open-source and freely available from <http://www.cibiv.at/software/mpboot>. It has been tested on Unix.

2 Installation

- 1) Download the executable version of MPBoot for your operating system if it is available (mpboot-XXX-OS.tar.gz or mpboot-XXX-OS.zip, where XXX is the current version number and OS the operating system) from <http://www.cibiv.at/software/mpboot>
- 2) Extract the files (e.g., with tar xvzf mpboot-XXX-OS.tar.gz under Unix). This should create a directory mpboot-XXX-OS.
- 3) You will find the executable in mpboot-XXX-OS/. This executable you should rename to mpboot and copy it to your system search path such that it is found by your system.

3 Compiling instruction

In case you want to compile the source code from scratch on a Linux machine

- 1) Download MPBoot source code from the website <http://www.cibiv.at/software/mpboot> (mpboot-XXX-Source.tar.gz or mpboot-XXX-Source.zip, where XXX is the current version number and OS the operating system)

- 2) Uncompress it and rename to **source**
- 3) Create folder **build** outside folder **source**
- 4) Change directory to **build**
- 5) Run **cmake** command:

```
cmake ../source -DIQTREE_FLAGS=sse4 -DCMAKE_C_COMPILER=clang
      -DCMAKE_CXX_COMPILER=clang++
```

- 6) Replace **sse4** by **avx** in above command if you decide to run MPBoot on AVX architecture
- 7) Run **make**
- 8) You will find the executable named **mpboot** (or **mpboot-avx**) once the **make** command is done.

4 Tutorial

4.1 Minimal command-line examples

(replace 'mpboot...' with actual path to executable)

- 1) Reconstruct maximum parsimony tree from a sequence alignment (example.phy):

```
mpboot -s example.phy
```

- 2) Reconstruct MP tree and assess branch supports with the MPBoot method (1000 replicates):

```
mpboot -s example.phy -bb 1000
```

4.2 All available options: run 'mpboot -h'

- mulhits Store multiple equally parsimonious trees per bootstrap replicate
- ratchet_iter <number> Number of non-ratchet iterations before each ratchet iteration (default: 1)
- ratchet_wgt <number> Weight to add to each site selected for perturbation during ratchet (default: 1)
- ratchet_percent <number> Percentage of informative sites selected for perturbation during ratchet (default: 50)
- ratchet_off Turn off ratchet, i.e. Only use tree perturbation
- spr_rad <number> Maximum radius of SPR (default: 6)
- cand_cutoff <#s> Use top #s percentile as cutoff for selecting bootstrap candidates (default: 10)
- opt_btree_off Turn off refinement step on the final bootstrap tree set
- nni_pars Hill-climb by NNI instead of SPR
- cost <file> Read <file> for the matrix of transition cost between character states.
Replace <file> by letter *e* for uniform cost.

4.3 Input/Output files description

- *Input alignment file*, e.g., example.phy, is of PHYLIP, FASTA, or NEXUS format.
- *Input cost file* (if using -cost option, e.g., -cost dna.cost) describes a square matrix. First line contains one integer implying the number n of character states. Each of the following n lines comprises n integers implying the cost for transitioning between character states. For DNA, set $n = 4$. For protein, set $s = 20$.
- *MPBoot report file*: example.phy.mpboot
- *Maximum-parsimony tree file*: example.phy.treefile
- *Split support values file*: example.phy.splits.nex
- *Consensus tree file*: example.phy.contree
- *Screen log file*: example.phy.log

Credits and Acknowledgement

Some parts of the code were taken from the following packages/libraries: Phylogenetic likelihood library (Flouri et al., 2014), TREE-PUZZLE (Schmidt et al., 2002), BIONJ (Gascuel, 1997), Nexus Class Library (Lewis, 2003), Eigen library (Guennebaud et al., 2010), SPRNG library (Mascagni and Srinivasan, 2000), Zlib library (<http://www.zlib.net>).