

---

## A guide to the stand-alone version of MLTreeMap

MLTreeMap is a software framework, designed for phylogenetic and functional analysis of metagenomic data. It searches for instances of marker genes on nucleotide sequences and deduces their most likely origin in a set of reference phylogenies. The current version of MLTreeMap can be downloaded and installed individually. MLTreeMap runs on Mac and Linux systems.

This guide can be downloaded from [http://mltreemap.org/treemap\\_cgi/show\\_download\\_page.pl](http://mltreemap.org/treemap_cgi/show_download_page.pl).

### A) Installation of MLTreeMap

#### Step 1

Unzip and unpack the file 'MLTreeMap\_package\_2\_051.tar.gz'.

#### Step 2

Enter the directory 'MLTreeMap\_package\_2\_051/install/'.

Type 'make'.

The make program now creates the data structure of MLTreeMap (to be found in 'MLTreeMap\_package\_2\_051/mltreemap\_2\_051/') and compiles most needed sub-programs (hmmalign, Genewise and RAxML).

#### Step 3

BLAST and Gblocks have to be added manually. For this enter the directory 'MLTreeMap\_package\_2\_051/install/sources/BLAST/'. Here you will find a collection of BLAST binaries. Choose the one appropriate for your system and copy it to the directory 'MLTreeMap\_package\_2\_051/mltreemap\_2\_051/sub\_binaries/'.

Repeat this for Gblocks  
(to be found in 'MLTreeMap\_package\_2\_051/install/sources/Gblocks/').

MLTreeMap is now ready to use. You can copy the directory mltreemap\_2\_051 to any place you like (as well as renaming it). The only dependency of MLTreeMap is that you must have Perl installed on your system.

### B) Usage of MLTreeMap

MLTreeMap has to be accessed on the command line.

An example for a valid input command is:  
`./mltreemap.pl -i example_input/rubisco.txt`

This will analyze the sequence in the file rubisco.txt and write the result to the output directory 'mltreemap\_2\_051/output/'.

---

Further (optional) input parameters are:

- b** number of bootstrap replicates (default: 0 i.e. no bootstrapping).
- c** usage a computer cluster (0 = no cluster (default), s = sun grid).
- f** RAxML algorithm (v = maximum likelihood (default), p = maximum parsimony).
- g** minimal sequence length after Gblocks (default: 50).
- l** long input files will be split into files of n sequences each (default: 2000).
- o** output directory (default: output/).
- s** minimum bitscore for the blast hits (default: 60).
- t** phylogenetic reference tree (p = MLTreeMap tree (default), g = GEBA tree).

### C) The MLTreeMap output

MLTreeMap searches for phylogenetic and functional marker genes. As soon as the results can be assigned to a specific marker gene, they are labelled accordingly by the first character of the output files.

Phylogenetic analysis:

- a** 16S rRNA reference tree
- b** 18S rRNA reference tree
- g** GEBA reference tree
- p** MLTreeMap reference tree

Functional analysis:

- c** Photolyase & cryptochrome
- d** Reverse dissimilatory sulfite reductase (DsrAB)
- e** HZO/HAO
- h** NifH (K02588)
- m** Methane & ammonia monooxygenase
- n** NifD (K02586)
- r** RuBisCo (COG1850)

Additionally, each output file after the RAxML step gets a header line, providing this information in words. In case of the RuBisCO example from above, the final output file would look as follows:

```
# Functional analysis, RuBisCO:
```

```
Placement weight 100%: Assignment of query to Acry 1067 (Acidiphilium cryptum) (6).
```