A geological timescale for bacterial evolution and oxygen adaptation – an mcmc-date tutorial

Lenard L. Szantho

1 Introduction

This tutorial walks you through the steps necessary to run `mcmc-date` on the datasets published in Davín et al. A geological timescale for bacterial evolution and oxygen adaptation (2024).

2 Setting up mcmc-date environment

Install Haskell and Cabal if not done already by following these guides:

- Haskell installation guide
- Cabal installation guide

Subsequently, clone the GitHub repository and prepare the Haskell environment as follows:

```
git clone https://github.com/dschrempf/mcmc-date.git
cd mcmc-date/tutorial_bacterial_rooting_goe
cabal update
cabal build
```

Verify the functionality of `mcmc-date`:

```
cabal run mcmc-date-run
```

3 (Optional) Inferring posterior distribution of species tree branch lengths

> **Optional**: Should one choose to bypass this step, the file `65genes_combined.treelist.tar.gz` is available for download from FigShare. Once downloaded, extract the file into the directory containing this tutorial. Subsequent commands will reference the path to the file `65genes_combined.treelist`; updates to the path might be necessary.

In the `data` folder, one finds the alignment `65genes_bac_and_organelles.phylip` and the corresponding inferred ML species tree `1007_mito_plastid.tree`. In order to date the tree, first we need to infer the posterior distribution of branch lengths of the supplied species tree topology. For this purpose, we will use Phylobayes-MPI which we instruct to keep the topology fixed while sampling branch lengths under the defined model.

First, if the tree is rooted, we have to unroot it using ete3 or a similar tool. This results in `data/1007_mito_plastid.tree.unrooted`

Now we can start 2 chains of Phylobayes-MPI with the following parameters (using LG exchangeability with only one category and G4):

```
mpirun -np 96 pb_mpi -lg -ncat 1 -dgam 4 \\
             -d 65genes_bac_and_organelles.phylip \\
             -T 1007_mito_plastid.tree.unrooted \\
             65genes_chain1
mpirun -np 96 pb_mpi -lg -ncat 1 -dgam 4 \\
             -d 65genes_bac_and_organelles.phylip \\
             -T 1007_mito_plastid.tree.unrooted \\
             65genes_chain2
```

After observing sufficient convergence, as described in Phylobayes MPI's tutorial and achieving a sample size of at least 10,000 iterations, we can stop the chains.

We can easily concatenate the treelists (containing the species tree branch length posterior distribution) that we need for the mcmc-date analysis:

```
paste -d "\n" 65genes_chain1.treelist 65genes_chain2.treelist > 65genes_combined.
   ↪treelist
```

4 Starting mcmc-date analysis

Create a directory containing the forthcoming `mcmc-date` analyses:

```
mkdir analyses
cd analyses
```

4.1 Preparing

Each analysis will use the same input rooted tree and posterior branch length distribution, hence it suffices to prepare them once and use them in the subsequent runs.

```
cabal run -- mcmc-date-run prepare --analysis-name "cyan28" \\
                   --rooted-tree "../data/1007_mito_plastid.tree" \\
                   --trees "../65genes_combined.treelist" \\
                   --likelihood-spec "SparseMultivariateNormal␣0.1"
cabal run -- mcmc-date-run prepare --analysis-name "cyan28_prioronly" \\
                   --rooted-tree "../data/1007_mito_plastid.tree" \\
                   --trees "../65genes_combined.treelist" \\
                   --likelihood-spec "NoLikelihood"
```

For `mcmc-date`'s `analyze` script to work properly, the analysis results must be in directories with names beginning with `results_`. Let us create these directories and symlink the corresponding files from the preparation step:

```
mkdir results_Fossils_cyan28
mkdir results_Fossils_cyan28_prioronly
mkdir results_XGBoost_cyan28
mkdir results_XGBoost_cyan28_prioronly

for d in results_*; do
    cd $d
    PREP=`echo $d | sed 's/results_//'`
    # symlinking the corresponding datafiles
    ln -s ../${PREP}.prepare.log
    ln -s ../${PREP}.data
    ln -s ../${PREP}.meantree
    cd ..
done
```

4.2 Fossils only

The file data/Fossils.csv contains solely fossil calibrations, without aerobicity data.

```
cd results_Fossils_cyan28
cabal run -- mcmc-date-run run  --analysis-name "Fossils_cyan28" \\
                --calibrations "csv␣../data/Fossils.csv" \\
                --ignore-problematic-calibrations \\
                --braces ../data/braces.json \\
                --relaxed-molecular-clock "UncorrelatedGamma" \\
                --likelihood-spec "SparseMultivariateNormal␣0.1"
cd ..

cd results_Fossils_cyan28_prioronly
cabal run -- mcmc-date-run run  --analysis-name "Fossils_cyan28_prioronly" \\
                --calibrations "csv␣../data/Fossils.csv" \\
                --ignore-problematic-calibrations \\
                --braces ../data/braces.json \\
                --relaxed-molecular-clock "UncorrelatedGamma" \\
                --likelihood-spec "NoLikelihood"
cd ..
```

4.3 XGBoost

The file data/XGBoost.csv contains both fossil and inferred aerobicity information.

```
cd results_XGBoost_cyan28
cabal run -- mcmc-date-run run  --analysis-name "XGBoost_cyan28" \\
                --calibrations "csv␣../data/XGBoost.csv" \\
                --ignore-problematic-calibrations \\
                --braces ../data/braces.json \\
                --relaxed-molecular-clock "UncorrelatedGamma" \\
                --likelihood-spec "SparseMultivariateNormal␣0.1"
cd ..

cd results_XGBoost_cyan28_prioronly
cabal run -- mcmc-date-run run  --analysis-name "XGBoost_cyan28_prioronly" \\
                --calibrations "csv␣../data/XGBoost.csv" \\
                --ignore-problematic-calibrations \\
                --braces ../data/braces.json \\
                --relaxed-molecular-clock "UncorrelatedGamma" \\
                --likelihood-spec "NoLikelihood"
cd ..
```

5 Analyse results

The analyze script under the scripts directory will go through all results_* directory's content and create summary statistics as described in mcmc-date's results tutorial

```
ln -s ../scripts/analyze
./analyze
```