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

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 if not done already:

```
apt install cabal-install clang lldb lld
```

Clone the github repository and set up Haskell:

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

Test if `mcmc-date` is working properly:

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

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

Optional: To skip this step decompress and use the `data/65genes_combined.treelist.tar.gz` file.

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 that we can 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, for example, ete3 or arbitrary 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 exchangebilities 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
```

Once we see sufficient convergence as described in Phylobayes MPI's tutorial and a sampling 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 our future mcmc-date analyses:

```
mkdir analyses
cd analyses
```

4.1 Preparing

Each analysis will use the same input rooted tree and posterior branch length distribution, thus it is enough to prepare them once and share with 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 analyse script to work properly, it is important that the analyses results are in directories with names beginning with `results_`. Let us create these 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 `data/Fossils.csv` calibrations are containing fossil calibrations only, no aerobicity information.

```
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 `data/XGBoost.csv` calibrations contain 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
```