BAli-Phy Tutorial

Benjamin Redelings

Table of Contents

1. Introduction

Before you start this tutorial, please download [http://www.bali-phy.org/download.php] and install bali-phy, following the installation instructions in the manual [http://www.bali-phy.org/README.html].

2. Setting up the `~/alignment_files` directory

Go to your home directory:

```
% cd ~
```

Make a directory called alignment_files inside it:

```
% mkdir alignment_files
```

Go into the `alignment_files` directory:

```
% cd alignment_files
```

Download the example alignment files:

```
% wget http://nucleus.biology.duke.edu/~bredelings/examples.tgz
```

Alternatively, you can use **curl**

```
% curl -O http://nucleus.biology.duke.edu/~bredelings/examples.tgz
```

Extract the compressed archive:

```
% tar -zxf examples.tgz
```

Take a look inside the `examples` directory:

```
% ls examples
```

Take a look at an input file:

```
% less examples/5S-rRNA/5d.fasta
```

Get some information about the alignment:

```
% alignment-info examples/5S-rRNA/5d.fasta
```

3. Command line options

What version of bali-phy are you running? When was it compiled? Which compiler? For what computer type?

```
% bali-phy -v
```

Look at the list of command line options:

```
% bali-phy --help
```

Look at them with the ability to scroll back:

```
% bali-phy --help | less
```

Some options have a short form which is a single letter:

```
% bali-phy -h | less
```

3.1. RNA

Analyze a data set, but don't begin MCMC. (This is useful to know if the analysis works, what model will be used, compute likelihoods, etc.)

```
% cd ~/alignment_files/examples
% bali-phy --test 5S-rRNA/5d.fasta
```

Finally, run an analysis! (This is just 50 iterations, so its not a real run.)

```
% bali-phy 5S-rRNA/5d.fasta --iterations=50
```

If you specify `--imodel=none`, then the alignment won't be estimated, and indels will be ignored (just like *MrBayes*).

```
% bali-phy 5S-rRNA/5d.fasta --iterations=50 --imodel=none
```

You can specify the alphabet, substitution model, insertion/deletion model, etc. Defaults are used if you don't specify.

```
% bali-phy 5S-rRNA/5d.fasta --iterations=50 --alphabet=DNA --smodel=TN --imodel=RS07
```

You can change this to the GTR, if you want:

```
% bali-phy 5S-rRNA/5d.fasta --iterations=50 --alphabet=DNA --smodel=GTR --imodel=RS07
```

You can add gamma[4]+INV rate heterogeneity:

```
% bali-phy 5S-rRNA/5d.fasta --iterations=50 --alphabet=DNA --smodel=GTR+gamma_inv[4] --imo
```

3.2. Amino Acids

When the data set contains amino acids, the default is amino acids:

```
% bali-phy EF-Tu/12d.fasta --iterations=50
```

3.3. Codons

What alphabet is used here? What substitution model?

```
% bali-phy HIV/chain-2005/env-clustal-codons.fasta --test
```

What happens when trying to use the Nielsen and Yang (1998) M0 model (e.g. dN/dS)?

```
% bali-phy HIV/chain-2005/env-clustal-codons.fasta --test --smodel=M0
```

The M0 model requires a codon alphabet:

```
% bali-phy HIV/chain-2005/env-clustal-codons.fasta --test --smodel=M0 --alphabet=Codons
```

The M0 model takes a *nucleotide* exchange model as a parameter. This parameter is optional, and the default is HKY, which you could specify as **`M0[HKY]`**. You can change this to be more flexible:

```
% bali-phy HIV/chain-2005/env-clustal-codons.fasta --test --smodel=M0[GTR] --alphabet=Codo
```

The M7 model is a mixture of M0 codon models:

```
% bali-phy Globins/bglobin.fasta --test --smodel=M7 --alphabet=Codons
```

The M7 model has parameters as well. Here are the defaults:

```
% bali-phy Globins/bglobin.fasta --test --smodel=M7[4,HKY,F61] --alphabet=Codons
```

It is possible to specify some of the parameters and leave others at their default value:

```
% bali-phy Globins/bglobin.fasta --test --smodel=M7[,TN] --alphabet=Codons
```

4. Output

See Section 6: Output [http://www.bali-phy.org/README.html#output] of the manual for more information about this section.

Try running an analysis with a few more iterations.

```
% bali-phy 5S-rRNA/25-muscle.fasta &
```

Run another copy of the same analysis:

```
% bali-phy 5S-rRNA/25-muscle.fasta &
```

You can take a look at your running jobs:

```
% jobs
```

4.1. Inspecting output files

Look at the directories that were created to store the output files:

```
% ls
% ls 25-muscle-1/
% ls 25-muscle-2/
```

See how many iterations have been completed so far:

```
% wc -l 25-muscle-1/C1.p 25-muscle-2/C1.p
```

Wait a second, and repeat the command.

```
% wc -l 25-muscle-1/C1.p 25-muscle-2/C1.p
```

See if you can determine the following information from the beginning of the C1.out file:

1. What command was run?
2. When was it run?
3. Which computer was it run on?
4. Which directory was it run in?
5. Which directory contains the output files?
6. What was the process id (PID) of the running bali-phy program?
7. What random seed was used?
8. What was the input file?
9. What alphabet was used to read in the sequence data?
10. What substitution model was used to analyze the sequence data?
11. What insertion/deletion model was used to analyze the sequence data?

```
% less 25-muscle-1/C1.out
```

Examine the file containing the sampled trees:

```
% less 25-muscle-1/C1.trees
```

Examine the file containing the sampled alignments:

```
% less 25-muscle-1/C1.P1.fastas
```

Examine the file containing the successive best alignment/tree pairs visited:

```
% less 25-muscle-1/C1.MAP
```

4.2. Summarizing the output

Try summarizing the sampled numerical parameters (e.g. not trees and alignments):

```
% statreport --help
% statreport 25-muscle-1/C1.p 25-muscle-2/C1.p > Report
% statreport 25-muscle-1/C1.p 25-muscle-2/C1.p --mean > Report
% statreport 25-muscle-1/C1.p 25-muscle-2/C1.p --mean --median > Report
% less Report
```

Now lets examine the summaries using a graphical program. If you are using Windows or Mac, run Tracer, and press the + button to add these files. What kind of ESS do you get? If you are using Linux, do

```
% tracer 25-muscle-1/C1.p 25-muscle-2/C1.p &
```

Now lets compute the consensus tree for these two runs:

```
% trees-consensus --help
% trees-consensus 25-muscle-1/C1.trees 25-muscle-2/C1.trees > c50.PP.tree
% trees-consensus 25-muscle-1/C1.trees 25-muscle-2/C1.trees --report=consensus > c50.PP.tr
% less consensus
% figtree c50.PP.tree &
```

Now lets see if there is evidence that the two runs have not converged yet.

```
% trees-bootstrap --help
% trees-bootstrap 25-muscle-1/C1.trees 25-muscle-2/C1.trees > partitions.bs
% less partitions.bs
```

4.3. Generating an HTML Report

Now lets use the analysis script to run all the summaries and make a report:

```
% bp-analyze.pl 25-muscle-1/ 25-muscle-2/
% firefox Results/index.html &
```

This PERL script runs *statreport* and *trees-consensus* for you. Take a look at what commands were run:

```
% less Results/bp-analyze.log
```

5. Starting and stopping the program

We didn't specify the number of iterations to run in the section above, so the two analyses will run for 100,000 iterations, or until you stop them yourself. See Section 10: Convergence and Mixing: Is it done yet? [http://www.bali-phy.org/README.html#mixing_and_convergence] of the manual for more information about when to stop an analysis.

In order to stop a running job, you need to kill it. One way of stopping bali-phy analyses is this:

```
% killall bali-phy
```

However, beware: if you are running multiple analyses, this will terminate all of them.

6. Multi-gene analyses

In this section we'll practice running analyses with multiple partitions. Dividing the data into multiple partitions is useful because different partitions can have different models, or can have different parameters for the same model. This is described in more detail in section 4.3 of the manual [http://www.bali-phy.org/README.html].

6.1. A simple multi-gene analysis

Let's look at a data set that is divided into three partitions:

```
% alignment-info ITS/ITS1-trimmed.fasta
% alignment-info ITS/5.8S.fasta
% alignment-info ITS/ITS2-trimmed.fasta
```

6.1.1. Running the analysis

We can run an analysis of this partitioned data simply by supplying a number of different alignment files as input to bali-phy. Let's run an analysis of these three alignment files:

```
% bali-phy ITS/ITS1-trimmed.fasta ITS/5.8S.fasta ITS/ITS2-trimmed.fasta --smodel=TN --imod
```

```
% bali-phy ITS/ITS1-trimmed.fasta ITS/5.8S.fasta ITS/ITS2-trimmed.fasta --smodel=TN --imodel=
```

You could leave off the **--smodel=TN --imodel=RS07** part of the command line:

```
% bali-phy ITS/ITS1-trimmed.fasta ITS/5.8S.fasta ITS/ITS2-trimmed.fasta &
```

This would give the same output, since TN and RS07 are the defaults.

6.1.2. What did the analysis do?

Now, lets look at sampled continuous parameters:

```
% statreport ITS1-trimmed-5.8S-ITS2-trimmed-1/C1.p | less
% tracer ITS1-trimmed-5.8S-ITS2-trimmed-1/C1.p &
```

You'll see that each partition has a TN (Tamura-Nei) substitution model, as well as an RS07 indel model. Each partition has its own copy of the TN parameters and the RS07 parameters.

6.1.3. Question

The partitions share a common tree shape, including the same relative branch lengths. However, the size of the tree for each partition is different. We scale the whole shared tree by mu1 in partition 1, mu2 in partition 2, etc. The mu parameters give the average branch length in that partition. Thus, partitions with a smaller mu value have slower evolution.

Do the different partitions of this data set have the same evolutionary rates? Do the different partitions of this data set have the same base frequencies?

6.2. Using different models in different partitions

6.2.1. Using different substitution models

Now lets try to specify different models for different partitions. Here we've used a command-line trick with curly braces {} to avoid typing some things multiple times.

```
% bali-phy ITS/{ITS1-trimmed,5.8S,ITS2-trimmed}.fasta --smodel=1:GTR --smodel=2:HKY --smodel=
```

We've also specified different substitution models for each partition. Take a look at the `C1.p` file for this analysis to see what parameters appear.

6.3. Using different indel models models

We can also specify different indel models for each partition:

```
% bali-phy ITS/{ITS1-trimmed,5.8S,ITS2-trimmed}.fasta --imodel=1:RS07 --imodel=2:none --imode
```

There are only two indel models: RS07, and none. Specifying **--imodel=none** removes the insertion-deletion model and parameters for a partition. It also disables alignment estimation for that partition.

6.4. Sharing model parameter between partitions

We can also specify that some partitions with the same model also share the same parameters for the model:

```
% bali-phy ITS/{ITS1-trimmed,5.8S,ITS2-trimmed}.fasta --smodel=1,3:GTR --imodel=1,3:RS07 --sm
```

This means that the information is pooled between the partitions to estimate the shared parameters.

6.5. Sharing substitution rates between partitions

We can also specify that some partitions with the same model also share the same parameters for the model:

```
% bali-phy ITS/{ITS1-trimmed,5.8S,ITS2-trimmed}.fasta --smodel=1,3:GTR --imodel=1,3:RS07 -
```

This means that the branch lengths for partitions 1 and 3 are the same, instead of being independently estimated.

7. Option files

You can collect command line options into a file for later use. Make a text file called `analysis1.script`:

```
align = ITS/ITS1-trimmed.fasta
align = ITS/5.8S.fasta
align = ITS/ITS2-trimmed.fasta
smodel = 1,3:TN+DP[3]
smodel = 2:TN
imodel = 2:none
same-scale = 1,3:mu1
```

You can run the analysis like this:

```
% bali-phy -c analysis1.script &
```

8. Dataset preparation

8.1. Splitting and Merging Alignments

BAli-Phy generally wants you to split concatenated gene regions in order to analyze them.

```
% cd ~/alignment-files/examples/ITS/
% alignment-cat -c1-223 ITS-region.fasta > 1.fasta
% alignment-cat -c224-379 ITS-region.fasta > 2.fasta
% alignment-cat -c378-551 ITS-region.fasta > 3.fasta
```

Later you might want to put them back together again:

```
% alignment-cat 1.fasta 2.fasta 3.fasta > 123.fasta
```

8.2. Shrinking the data set

You might want to reduce the number of taxa while attempting to preserve phylogenetic diversity:

```
% alignment-thin --down-to=30 ITS-region.fasta > ITS-region-thinned.fasta
```

You can specify that certain sequences should not be removed:

```
% alignment-thin --down-to=30 --keep=Csaxicola420 ITS-region.fasta > ITS-region-thinned.fa
```

8.3. Cleaning the data set

Keep only columns with a minimum number of residues:

```
% alignment-thin --min-letters=5 ITS-region.fasta > ITS-region-censored.fasta
```

Keep only sequences that are not too short:

```
% alignment-thin --longer-than=250 ITS-region.fasta > ITS-region-long.fasta
```

Remove 10 sequences with the smallest number of conserved residues:

```
% alignment-thin --remove-crazy=10 ITS-region.fasta > ITS-region-sane.fasta
```