

BAl i-Phy Tutorial

Benjamin Redelings

Table of Contents

- [1. Introduction](#)
- [2. Setting up the ~/Work directory](#)
- [3. Command line options](#)
 - [3.1. RNA](#)
 - [3.2. Amino Acids](#)
 - [3.3. Codons](#)
- [4. Output](#)
 - [4.1. Inspecting output files](#)
 - [4.2. Summarizing the output](#)
- [5. Multi-gene analyses](#)

1. Introduction

This tutorial contains step-by-step commands, assuming that you have already installed bali-phy in ~/local/. The [manual](#) contains more detailed explanations for many things, so you might want to refer to it during the tutorial.

2. Setting up the ~/Work directory

Go to your home directory:

```
% cd ~
```

Make a directory called Work inside it:

```
% mkdir Work
```

Go into the Work directory:

```
% cd Work
```

Make a shortcut (a symbolic link) to the examples directory:

```
% ln -s ~/local/share/bali-phy/examples examples
```

Take a look inside the examples directory:

```
% ls examples
```

Take a look at an input file:

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

Take a look at an input file:

```
% 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.)

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

Finally, run an analysis! (This is just 10 iterations, so its not a real run.)

```
% bali-phy examples/5S-rRNA/5d.fasta --iterations=10
```

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

```
% bali-phy examples/5S-rRNA/5d.fasta --iterations=10 --imodel=none
```

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

```
% bali-phy examples/5S-rRNA/5d.fasta --iterations=10 --alphabet=DNA --smodel=TN --imodel=RS07
```

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

```
% bali-phy examples/5S-rRNA/5d.fasta --iterations=10 --alphabet=DNA --smodel=GTR --imodel=RS07
```

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

```
% bali-phy examples/5S-rRNA/5d.fasta --iterations=10 --alphabet=DNA --smodel=GTR+Rates.Gamma[4]+INV --imodel=RS07
```

3.2. Amino Acids

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

```
% bali-phy examples/EF-Tu/12d.fasta --iterations=10
```

3.3. Codons

What alphabet is used here? What substitution model?

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

What happens when trying to use the M0 model (e.g. positive selection)?

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

The M0 model requires a codon alphabet:

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

4. Output

See [Section 6: Output](#) of the manual for more information about this section.

Try running an analysis with a few more iterations. The & makes the command run in the background, so that you can run other commands before it finishes.

```
% balı-phy examples/5S-rRNA/25-muscle.fasta --iterations=200 &
```

Run another copy of the same analysis:

```
% balı-phy examples/5S-rRNA/25-muscle.fasta --iterations=200 &
```

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: What command was run? When was it run? Which computer was it run on? Which directory was it run in? Which directory contains the output files? What was the process id (PID) of the running balı-phy program? What random seed was used? What was the input file? What alphabet was used to read in the file? What substitution model was used to analyze the file? What insertion/deletion model was used to analyze the file?

```
% 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 let's examine the summaries using a graphical program. If you are using Windows or Mac, run Tracer, and press the \pm 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 let's 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.tree
% less consensus
% figtree c50.PP.tree &
```

Now let's 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
```

Now let's 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 &
```

5. Multi-gene analyses

This is described in more detail in section 4.3 of the [manual](#).

Download HIV env and pol alignments with the same sequence names:

```
% wget http://nucleus.biology.duke.edu/~bredelings/alignment_files/env-common.fasta
% wget http://nucleus.biology.duke.edu/~bredelings/alignment_files/pol-common.fasta
```

Try to analyze these genes jointly under a codon model:

```
% bali-phy --test --alphabet Codons env-common.fasta pol-common.fasta --smodel M0
```

Take a look at the lengths of these files, to see whether the sequence alignment lengths are multiples of 3:

```
% alignment-info pol-common.fasta
% alignment-info env-common.fasta
```

Now edit the fasta files so that the lengths are a multiple of three:

```
% alignment-cat -c1-951 env-common.fasta > env-common2.fasta
% alignment-cat -c1-1068 pol-common.fasta > pol-common2.fasta
```

Try running bali-phy again:

```
% bali-phy --test --alphabet Codons env-common2.fasta pol-common2.fasta --smodel M0
```

Now open the fasta files in an editor and replace all nucleotide ambiguity codes K and M with N. This time the run should succeed:

```
% bali-phy --test --alphabet Codons env-common2.fasta pol-common2.fasta --smodel M0
```

By default, the M0 substitution model will be selected for both genes. However, each gene will have a separate instance of the model with its own gene-specific values for the model parameters. Try specifying that both genes share one instance of the model parameters:

```
% bali-phy --iterations=10 --alphabet Codons env-common2.fasta pol-common2.fasta --smodel=1,2:M0
```

Now try reading one gene as Codons, and the other as DNA:

```
% bali-phy --iterations=10 --alphabet=1:Codons --alphabet=2:DNA env-common2.fasta pol-common2.fasta
```