BAli-Phy Tutorial (for version 4.3)

Benjamin Redelings

Table of Contents

1. Introduction
2. Setting up the `~/alignment_files` directory
3. Command line options
4. Help
5. Analysis 1: 5S rRNA -- free alignment versus fixed alignment
    5.1. Free alignment
    5.2. Fixed alignment
6. Analysis 2: ITS sequences -- multi-gene analysis
    6.1. Questions about parameter differences between genes
7. Analysis 3: Exons and Introns
    7.1. Extracting parts of a gene
    7.2. The ASRV.Free model
    7.3. Fixing the alignment of some partitions
    7.4. Linking parameters between partitions
    7.5. Character sets
    7.6. Option files
    7.7. Questions about intron alignments
    7.8. Exons as codons + introns as nucleotides
    7.9. Exons as amino acids + introns as nucleotides
8. Analysis 4: ITS sequences - with a better model
9. Analysis 5: Mx genes -- codon models and selection
    9.1. One ω for the whole sequence
    9.2. Different ω values among sites
    9.3. Testing for positively selected sites
    9.4. Different ω values among branches
    9.5. Testing for a difference between branch groups
    9.6. Testing for selection at some sites on foreground branches
    9.7. A historical codon model
10. Complex substitution models
    10.1. Defaults
    10.2. Rate variation
    10.3. Heterotachy
    10.4. Fixing parameter values
    10.5. Priors
11. Specifying the model for each partition
    11.1. Using different substitution models

11.2. Disabling alignment estimation for some partitions
11.3. Sharing model parameters between partitions
11.4. Sharing substitution rates between partitions
12. Dataset preparation
12.1. Splitting and merging Alignments
12.2. Shrinking the data set
12.3. Cleaning the data set

1. Introduction

This tutorial offers a quick overview of bali-phy with an emphasis on concrete analysis of real datasets. Other documentation includes:

- The User Guide is much more complete.
- Run *`tool`* `--help` to see command line options for *`tool`*.
- Run `man` *`tool`* to see the manual page for *`tool`*.
- Manual pages for bali-phy and tools are also available online.

Before you start this tutorial, please download and install bali-phy, following the installation instructions in the User Guide.

The shell commands below work in bash and zsh on Linux and macOS. On Windows, use WSL for this tutorial; see the User Guide's installation instructions. The `%` marks the shell prompt and is not part of the command.

The analyses below are exercises, not a guarantee of convergence. You can run `bpy-summarize` while the analyses continue to check their progress. Before using results for inference, assess convergence and effective sample sizes as described in the User Guide.

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 https://www.bali-phy.org/examples.tgz
```

Alternatively, you can use `curl`

```
% curl -O https://www.bali-phy.org/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 (you can press 'q' to exit 'less'):

```
% 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
% bali-phy --version
```

Look at the list of command line options:

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

By default, not all the options are shown. To see more options, try:

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

All bali-phy tools also take `--help`.

```
% alignment-thin --help | less
% alignment-cat --help  | less
```

4. Help

You can also get help on models, distributions, functions, and command-line options:

```
% bali-phy help TN93           | less        # Tamura-Nei (1993) model
% bali-phy help Normal         | less        # Normal distribution
% bali-phy help quantile       | less        # quantile function
```

You can find out which models, distributions, and functions are available:

```
% bali-phy help models         | less        # What models are available?
% bali-phy help distributions  | less        # What distributions are available?
% bali-phy help functions      | less        # What functions are available?
```

You can get extended help on each command line option:

```
% bali-phy help iterations     | less        # --iterations=<num> command line option
% bali-phy help alphabet       | less        # --alphabet=<a> command line option
```

5. Analysis 1: 5S rRNA -- free alignment versus fixed alignment

Let's estimate a phylogeny on a very small data set under both a free alignment and a fixed alignment to see how the alignment affects our confidence in the estimated topology.

5.1. Free alignment

Let's get started by going to the examples directory:

```
% cd ~/alignment_files/examples/5S-rRNA
```

Now let's start a run.

```
% bali-phy 5d-clustalw.fasta --smodel 'GTR +> ASRV.Gamma(4) +> Inv' --name 5d-freeA &
```

You have started the first copy of the analysis. BAli-Phy will create a directory called `5d-freeA-1` to hold output files from this analysis, as directed by the `--name 5d-freeA` option.

Let's run another copy of the same analysis:

```
% bali-phy 5d-clustalw.fasta --smodel 'GTR +> ASRV.Gamma(4) +> Inv' --name 5d-freeA &
```

This command will create another directory called `5d-freeA-2` for its own output files. This additional run will take advantage of a second processor. It will also allow us to determine if the two runs have performed enough iterations to agree.

You can take a look at jobs running in the background. There should be two bali-phy jobs running:

```
% jobs
```

The previous command only shows jobs started by the shell you are typing in. You can also examine all processes running on your computer. bali-phy should show up near the top since it is using a lot of the CPU:

```
% top          # press 'q' to exit top
```

In order to see how many iterations have completed, check the number of lines in the log files:

```
% wc -l 5d-freeA-*/C1.log
```

To summarize the results generated so far, run the summary script and then view the results in a web browser:

```
% bpy-summarize 5d-freeA-1/ 5d-freeA-2/
```

Open `Results/index.html` in your web browser.

For this exercise, collect about 2000 iterations from each run, then summarize the latest results:

```
% wc -l 5d-freeA-*/C1.log
% bpy-summarize 5d-freeA-1/ 5d-freeA-2/
% mv Results 5d-freeA.Results
```

Open `5d-freeA.Results/index.html` in your web browser.

Now let's kill the bali-phy processes:

```
% top         # use top to find the PID (process id) for each of the bali-phy processes
% kill pid1
% kill pid2
% top         # check that the jobs have stopped.
```

If FigTree is installed, you can also examine the tree with it:

```
% figtree 5d-freeA.Results/c50.PP.tree &
```

1. When it says "Please select a name for these values", enter `PP`.
2. Then, enable "Branch Labels".
3. Under "Branch Labels / Display", select "PP".
4. You can increase the font size to make the figure more readable.
5. You can also increase the font size of the tip labels.

5.1.1. Parameter questions:

- What is the meaning of `Inv:pInv` parameter? (Run `bali-phy help Inv`)
- What is the prior distribution on `Inv:pInv`? (See *Model and priors / Substitution model*)
- What is the posterior median for `Inv:pInv`? (See *Scalar Variables*)

5.1.2. Alignment questions:

- Which part of the alignment is most uncertain? (See *Alignment Distribution: follow an AU link for a consensus alignment*)

5.1.3. Tree questions (free alignment):

- How many internal branches are in the 50% consensus tree?
- What is the posterior probability for each of these internal branches?

5.2. Fixed alignment

```
% cd ~/alignment_files/examples/5S-rRNA
% bali-phy 5d-clustalw.fasta -S 'GTR +> ASRV.Gamma(4) +> Inv' -I none -n 5d-fixedA &
% bali-phy 5d-clustalw.fasta -S 'GTR +> ASRV.Gamma(4) +> Inv' -I none -n 5d-fixedA &
```

The `-I none` is a short form of `--imodel=none`, where *`imodel`* means the insertion-deletion model. When there's no model of insertions and deletions, then the alignment must be kept fixed.

Summarize the results while the analyses run. For this exercise, collect about 2000 iterations from each run before comparing the fixed and free alignments:

```
% wc -l 5d-fixedA-*/C1.log
% bpy-summarize 5d-fixedA-1/ 5d-fixedA-2
```

Examine the majority consensus tree for the fixed alignment analysis:

```
% figtree Results/c50.PP.tree &
```

5.2.1. Tree questions (fixed alignment):

- How many internal branches are in the 50% consensus tree?
- What is the posterior probability for each of these internal branches?
- For this data set, how does fixing the alignment affect our confidence in the tree? Why might this be?

Stop these two runs when you have finished, using their process IDs as in the free-alignment example.

6. Analysis 2: ITS sequences -- multi-gene analysis

Let's do an analysis of intergenic transcribed spacer (ITS) genes from 20 specimens of lichenized fungi (Gaya et al, 2011 analyzes 68 specimens). The ITS genes contain a lot of variation, but are hard to align. This analysis will show how to estimate the alignment, phylogeny, and evolutionary parameters using MCMC. Averaging over all alignments makes it safe to use the ITS region without throwing out ambiguously-aligned regions.

This data set is divided into three gene regions, or partitions. It is assumed that all genes evolve on the same tree, but may have different rates and evolutionary parameters. Let's look at the sequences. How long are they?

```
% cd ~/alignment_files/examples/ITS
% alignment-info ITS1.fasta
% alignment-info 5.8S.fasta
% alignment-info ITS2.fasta
```

By default, each gene gets a default substitution model based on whether it contains DNA/RNA or amino acids. By running bali-phy with the `--test` option, we can reveal what substitution models and priors will be used, without actually starting a run.

```
% bali-phy ITS1.fasta 5.8S.fasta ITS2.fasta --test
```

Let's run two copies of this analysis:

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

Since we did not specify a name for the analysis, bali-phy creates the name `ITS1-5.8S-ITS2` from the input file names. Let's analyze the results, even if the analysis has not yet converged:

```
% bpy-summarize ITS1-5.8S-ITS2-1/ ITS1-5.8S-ITS2-2/
```

Open `Results/index.html` in your web browser.

The optional program Tracer graphically displays the posterior probability distribution for each parameter.

```
% tracer ITS1-5.8S-ITS2-1/C1.log ITS1-5.8S-ITS2-2/C1.log &
```

If you are using Windows or Mac, first run Tracer, and then press the **+** button to add the files.

> ### 6.1. Questions about parameter differences between genes
>
> These genes have different evolutionary processes. How does the evolutionary process for these genes differ in:
>
> 1. substitution rate? (`scale1`, `scale2`, ...)

> 2. insertion-deletion rate? (`I1/RS07:rate`, `I2/RS07:rate`, ...)
> 3. nucleotide frequencies? (`S1/TN93:pi[A]`, `S1/TN93:pi[C]`, ...)
> 4. number of indels? (`#indels`)

7. Analysis 3: Exons and Introns

Let's look at a data set containing both exons and introns. The pattern of dividing a sequence into high-rate and low-rate regions also applies to stems and loops in RNA sequences. It is helpful to allow different insertion-deletion rates in conserved and non-conserved regions of a gene.

7.1. Extracting parts of a gene

First, let's split the gene into separate FASTA files for each intron and exon:

```
% cd ~/alignment_files/examples/ferns/
% alignment-cat -c 5-8       -e cleaned.fasta > exon1.fasta
% alignment-cat -c 9-453     -e cleaned.fasta > intron1.fasta
% alignment-cat -c 454-596   -e cleaned.fasta > exon2.fasta
% alignment-cat -c 597-864   -e cleaned.fasta > intron2.fasta
% alignment-cat -c 865-948   -e cleaned.fasta > exon3.fasta
% alignment-cat -c 949-1328  -e cleaned.fasta > intron3.fasta
% alignment-cat -c 1329-1393 -e cleaned.fasta > exon4.fasta
```

Now, let's put the pieces back together in case we need the complete gene in the future:

```
% alignment-cat exon1.fasta intron1.fasta exon2.fasta intron2.fasta exon3.fasta
  intron3.fasta exon4.fasta > combined.fasta
```

It is also possible to refer to character sets without creating a file for each one, as mentioned below.

7.2. The ASRV.Free model

We will add `-S 'GTR +> ASRV.Free(n=4)'` to use the free-rates model of substitution with 4 categories:

```
% bali-phy exon1.fasta intron1.fasta exon2.fasta intron2.fasta exon3.fasta intron3.fasta
  exon4.fasta -S 'GTR +> ASRV.Free(n=4)' --test
```

This model is more general than ASRV.Gamma, since it estimates 4 free rates and the frequencies of the 4 rates. These 8 parameters have 6 degrees of freedom, which is a lot more than the 1 degree of freedom for the `ASRV.Gamma` model. Furthermore, each partition by default gets a separate copy of the model, which leads to 42 degrees of freedom just for the `ASRV.Free` part of the model!

7.3. Fixing the alignment of some partitions

We will now add `-I 1,3,5,7:none` to disable alignment estimation, but only for the exons:

```
% bali-phy exon1.fasta intron1.fasta exon2.fasta intron2.fasta exon3.fasta intron3.fasta
  exon4.fasta -S 'GTR +> ASRV.Free(n=4)' -I 1,3,5,7:none --test
```

7.4. Linking parameters between partitions

We can reduce the number of parameters by using one set of parameters for the exons, and one set for the introns. This is called *linking* the models.

```
% bali-phy exon1.fasta intron1.fasta exon2.fasta intron2.fasta exon3.fasta intron3.fasta
  exon4.fasta -I 1,3,5,7:none -S '1,3,5,7:GTR +> ASRV.Free(n=4)' -S '2,4,6:GTR +>
  ASRV.Free(n=4)' --test
```

We would also like to share the indel model between introns:

```
% bali-phy exon1.fasta intron1.fasta exon2.fasta intron2.fasta exon3.fasta intron3.fasta
  exon4.fasta -I 1,3,5,7:none -S '1,3,5,7:GTR +> ASRV.Free(n=4)' -S '2,4,6:GTR +>
  ASRV.Free(n=4)' -I 2,4,6: --test
```

7.5. Character sets

Instead of creating separate files for each intron and exon, we can also refer to character sets of the combined file:

```
% bali-phy cleaned.fasta:5-8 cleaned.fasta:454-596 cleaned.fasta:865-948
  cleaned.fasta:1329-1393 cleaned.fasta:9-453 cleaned.fasta:597-864
  cleaned.fasta:949-1328 -I 1,2,3,4:none -S '1,2,3,4:GTR +> ASRV.Free(n=4)' -S
  '5,6,7:GTR +> ASRV.Free(n=4)' -I 5,6,7: --test
```

However, the command line is getting so long that it is difficult to manage.

7.6. Option files

Instead of writing all options on the command line, we can put some of them into a text file that is easier to manage and edit. Let's make a text file called `options1.txt`:

Example 1. `options1.txt`

```
# exon1
:align cleaned.fasta:5-8
# intron1
:align cleaned.fasta:9-453
# exon2
:align cleaned.fasta:454-596
# intron2
:align cleaned.fasta:597-864
```

```
# exon3
:align cleaned.fasta:865-948
# intron3
:align cleaned.fasta:949-1328
# exon4
:align cleaned.fasta:1329-1393

# exons
:smodel 1,3,5,7:GTR +> ASRV.Free(n=4)
:imodel 1,3,5,7:none

# introns
:smodel 2,4,6:GTR +> ASRV.Free(n=4)
:imodel 2,4,6:
```

We can then run

```
% bali-phy -c options1.txt --test
```

Let's run two copies of this analysis:

```
% bali-phy -c options1.txt &
% bali-phy -c options1.txt &
```

> ### 7.7. Questions about intron alignments
>
> - How do bali-phy alignments of the introns differ from muscle and mafft alignments of the introns?
> - How much certainty or uncertainty do the intron alignments have?
> - How do the evolutionary rates of the exons compare to the evolutionary rates of the introns?
> - Would it be reasonable to link the intron rates? How about the exon rates?

7.8. Exons as codons + introns as nucleotides

Another thing that we can do is to use a codon model on the combined exons, while using a nucleotide model on the introns. In order to load nucleotide sequences as codons, three conditions must be met:

1. **Sequence lengths must be multiples of three.**

   `AAA ---` TTT is OK

   `AAA ---` T`--` is not OK.

2. **Gaps must be codon-aligned.**

   `AAA ---` TTT is OK

`AA- --A TT-` is not OK.

3. **The sequences must be in-frame and not have stop codons.**

   `ATG AAT AAA` is OK, because it translates to `MNK`.

   `AAT GAA TAA` is not OK, because it translates to `NE*`, where * is a stop codon.

Note that spaces in FASTA files are allowed, but not required.

Let's first extract the coding data:

```
% alignment-cat exon{1,2,3,4}.fasta -c2-295 > coding.fasta     # Concatenate exons and
chop off partial codons at beginning and end
% sed '/^>/!s/-/N/g' coding.fasta > coding-with-N.fasta        # Replace gaps in
sequences, preserving FASTA names
% mv coding-with-N.fasta coding.fasta
```

We can then construct an option file `options2.txt`:

Example 2. `options2.txt`

```
:align intron1.fasta
:align intron2.fasta
:align intron3.fasta
:align coding.fasta

:smodel 1,2,3:GTR +> ASRV.Free(n=4)
:imodel 1,2,3:RS07

:smodel 4:GY94
#:smodel 4:GY94_ext(GTR_sym)     # A GTR-based version of GY94 a.k.a. M0.
#:smodel 4:GTR +> FMutSel0
#:smodel 4:|w:FMutSel0(omega=w)| +> M3
:imodel 4:none
```

We can choose any one of the three codon models for the coding sequences in the 4th partition.

```
% bali-phy -c options2.txt --test
```

Try changing the model to `GTR +> FMutSel0`. Does the number of parameters increase or decrease?

7.9. Exons as amino acids + introns as nucleotides

We can also translate the coding sequence into a protein sequence, while using a nucleotide model on the introns. Let's first extract the coding data:

```
% alignment-translate < coding.fasta > protein.fasta
```

We can then construct an option file `options3.txt`:

Example 3. `options3.txt`

```
:align intron1.fasta
:align intron2.fasta
:align intron3.fasta
:align protein.fasta

:smodel 1,2,3:GTR +> ASRV.Free(n=4)
:imodel 1,2,3:RS07

:smodel 4:LG +> ASRV.LogNormal(n=4)
:imodel 4:none
```

```
% bali-phy -c options3.txt --test
```

8. Analysis 4: ITS sequences - with a better model

Let's revisit the ITS sequences used in analysis 2. Now that you've been exposed to different models in different partitions, let's use a more complex substitution model for the ITS partitions (`--smodel '1,3:TN93 +> ASRV.Free(n=3)'`). Note that this also links the substitution models for the ITS partitions.

We can also fix the alignment for the 5.8S partition, since it has almost no indels. Return to the ITS directory and save the following options as `options4.txt`:

```
% cd ~/alignment_files/examples/ITS
```

Example 4. `options4.txt`

```
:align ITS1.fasta
:align 5.8S.fasta
:align ITS2.fasta

:smodel 1,3:TN93 +> ASRV.Free(n=3)
:smodel 2:TN93

:imodel 2:none

:scale 1,3:
```

```
% bali-phy -c options4.txt --test
```

This file additionally links the *scale* for the two ITS partitions (`--scale=1,3:`). This forces them to share the same evolutionary rate, and allows more precise estimates of the shared scale. You can run the analysis with unlinked scales by removing this option in order to determine if (i) the scales are similar enough to share and (ii) the scales have a high enough variance that they need to be linked.

9. Analysis 5: Mx genes -- codon models and selection

We will analyze myxovirus resistance (Mx) genes from ten mammals and two birds, using sequences from Álvarez-Carretero et al. (2023). See `Mx/README.md` in the examples for the citation, provenance, and preprocessing details. The terminal stop codons have been removed; the starting alignment contains 663 codon columns.

```
% cd ~/alignment_files/examples/Mx
```

These examples jointly estimate the alignment and evolutionary parameters. This reduces false signals of positive selection caused by misaligned codons and can improve alignment accuracy; site models also let alignment inference account for differences in ω among sites. See Redelings (2014). BAli-Phy also supports fixed-alignment analyses: add `-I none` to any BAli-Phy command below to hold the input alignment fixed.

Run each analysis twice and check convergence before interpreting results. Commands with `--test` only check the setup. After collecting samples, summarize with `bpy-summarize` and open the named report's `index.html`. Use the actual run suffixes if previous runs exist; move aside an old report before reusing its name. See the User Guide for convergence, burn-in, and effective sample sizes. Stop completed runs as in Analysis 1.

9.1. One ω for the whole sequence

Start with one ω shared across sites and branches. GTR describes nucleotide mutation; MNM allows changes of more than one nucleotide within a codon; MutSelAA estimates amino-acid preferences shared across sites; and dNdS adds the ω multiplier. Here ω describes nonsynonymous change beyond the effect of these preferences, not a raw ratio of observed changes.

```
% bali-phy Mx_aln.nonstop.fasta -A Codons -S 'GTR +> MNM +> MutSelAA +> dNdS' --name Mx-
  single-freeA &
% bali-phy Mx_aln.nonstop.fasta -A Codons -S 'GTR +> MNM +> MutSelAA +> dNdS' --name Mx-
  single-freeA &
% bpy-summarize Mx-single-freeA-1/ Mx-single-freeA-2/
% mv Results Mx-single-freeA.Results
```

Inspect the inferred alignment and the ω estimate and credible interval. A single ω averages across sites and cannot identify selected sites. Each subsequent model estimates its own alignment jointly with its parameters; do not fix the alignment to this first estimate. For the remaining analyses, run each launch command twice before summarizing.

9.2. Different ω values among sites

M8a allows a distribution of ω values below one and an additional category with ω equal to one. It describes variation among sites without allowing a positive-selection category. The expression `|w:...|` builds the codon model for each supplied ω; the mutation parameters and amino-acid preferences are shared across categories.

```
% bali-phy Mx_aln.nonstop.fasta -A Codons -S '|w:GTR +> MNM +> MutSelAA +>
  dNdS(omega=w)| +> M8a' --name Mx-M8a-freeA &
% bpy-summarize Mx-M8a-freeA-1/ Mx-M8a-freeA-2/
% mv Results Mx-M8a-freeA.Results
```

Inspect the site-model parameters and category proportions. M8a describes variation in ω; it does not test for positive selection.

9.3. Testing for positively selected sites

M8a_test compares M8a with M8, whose additional category has ω above one. Both hypotheses are included in one analysis: you do not need to fit M8a and M8 separately to perform this test.

```
% bali-phy Mx_aln.nonstop.fasta -A Codons -S '|w:GTR +> MNM +> MutSelAA +>
  dNdS(omega=w)| +> M8a_test' --name Mx-M8a-test-freeA &
% bpy-summarize Mx-M8a-test-freeA-1/ Mx-M8a-test-freeA-2/
% mv Results Mx-M8a-test-freeA.Results
```

Under `Positive selection`, read `Support for positive selection in the model`. See the User Guide for interpreting posterior support, prior odds, and Bayes factors.

In the alignment viewer, select `posSelection`. `Model-averaged` probabilities include both hypotheses; `Conditional on positiveSelectionInModel` assumes selection is present. High conditional probabilities alone do not establish support for selection.

Probabilities follow individual codons across sampled alignments, so codons displayed in the same column can have different probabilities. Identify codons by sequence and ungapped position. See the User Guide for details.

9.4. Different ω values among branches

`Mx_foreground.tree` labels the terminal chicken and duck branches with `:[&foreground=1]`; unlabelled branches use category 0. BranchModel assigns one ω to each group, shared across sites. Fixing the topology preserves these groups; branch lengths and alignment are still estimated. The preceding analyses also estimate topology.

```
% bali-phy Mx_aln.nonstop.fasta -A Codons --fix topology=Mx_foreground.tree -S '|w:GTR
  +> MNM +> MutSelAA +> dNdS(omega=w)| +> BranchModel' --name Mx-branch-freeA &
```

```
% bpy-summarize Mx-branch-freeA-1/ Mx-branch-freeA-2/
% mv Results Mx-branch-freeA.Results
```

Under `Branch models`, compare the category ω estimates. Different values need not imply positive selection; both can be below one.

9.5. Testing for a difference between branch groups

For BranchModel_test, use `Mx_branch_test.tree`. Each bird terminal branch has `:[&foreground={0,1}]`. The entries specify its category under each hypothesis: hypothesis 0 uses category 0 everywhere; hypothesis 1 assigns a shared category 1 to chicken and duck. Other branches stay in category 0 under both hypotheses.

```
% bali-phy Mx_aln.nonstop.fasta -A Codons --fix topology=Mx_branch_test.tree -S '|w:GTR
  +> MNM +> MutSelAA +> dNdS(omega=w)| +> BranchModel_test' --name Mx-branch-test-freeA
  &
% bpy-summarize Mx-branch-test-freeA-1/ Mx-branch-test-freeA-2/
% mv Results Mx-branch-test-freeA.Results
```

Under `Branch models`, read `Hypothesis support` and ω estimates for each hypothesis. This tests for different ω values between groups, not specifically for ω above one.

9.6. Testing for selection at some sites on foreground branches

BranchSite is a test, despite having no `_test` suffix. It allows some sites to have ω above one on foreground branches while their background ω remains at or below one, and compares this with a null hypothesis that fixes their foreground ω to one. Use the scalar `foreground=1` labels in `Mx_foreground.tree`, rather than the hypothesis lists used for BranchModel_test.

```
% bali-phy Mx_aln.nonstop.fasta -A Codons --fix topology=Mx_foreground.tree -S '|w:GTR
  +> MNM +> MutSelAA +> dNdS(omega=w)| +> BranchSite' --name Mx-branch-site-freeA &
% bpy-summarize Mx-branch-site-freeA-1/ Mx-branch-site-freeA-2/
% mv Results Mx-branch-site-freeA.Results
```

Read `Support for positive selection in the model`. In the alignment viewer, select `foreground-posSelection` or `foreground-dNdS`; the corresponding `background-` properties describe other branches. Distinguish model-averaged from selection-conditional probabilities as in the site test. Choose foreground branches from the biological question before examining selection results.

9.7. A historical codon model

For comparison with older analyses, the Goldman–Yang model with F1x4 frequencies can be specified as follows. Here `--test` only displays the setup:

```
% bali-phy Mx_aln.nonstop.fasta -A Codons --test -S 'GY94(pi=F1x4)'
```

This historical model does not include the MNM and amino-acid preference components used above. See the User Guide's background-model discussion for the reasons behind the choices in the worked examples.

10. Complex substitution models

While those analyses are running, let's look at how to specify more complex substitution models in bali-phy.

10.1. Defaults

When you don't specify values for parameters like *imodel*, bali-phy uses sensible defaults. For example, these two commands are equivalent:

```
% cd ~/alignment_files/examples/
% bali-phy 5S-rRNA/25-muscle.fasta --test
% bali-phy 5S-rRNA/25-muscle.fasta --test --alphabet=RNA --smodel=TN93 --imodel=RS07
```

You can change the substitution model from the Tamura-Nei model to the General Time-Reversible model:

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

Here the `-S GTR` is a short form of `--smodel=GTR`, where *smodel* means the substitution model.

10.2. Rate variation

You can also allow different sites to evolve at 5 different rates using the ASRV.Gamma(4) + Inv model of rate heterogeneity:

```
% bali-phy 5S-rRNA/25-muscle.fasta --test -S 'GTR +> ASRV.Gamma(4) +> Inv'
```

You can also use ASRV.LogNormal instead of ASRV.Gamma. ASRV.LogNormal+Inv is sometimes better behaved than ASRV.Gamma+Inv, because the smallest rate bin under ASRV.LogNormal is not quite as close to 0.

```
% bali-phy 5S-rRNA/25-muscle.fasta --test -S 'GTR +> ASRV.LogNormal(4) +> Inv'
```

You can allow 5 different rates that are all independently estimated:

```
% bali-phy 5S-rRNA/25-muscle.fasta --test -S 'GTR +> ASRV.Free(n=5)'
```

10.3. Heterotachy

Heterotachy occurs when the evolutionary rate changes across the tree. One version of heterotachy is the Tuffley-Steel (1998) model, where each state alternates between an ON and OFF state.

```
% bali-phy 5S-rRNA/25-muscle.fasta --test -S 'GTR +> Covarion.TuffleySteel98'
```

The Huelsenbeck (2002) model allows combining rate variation with Tuffley-Steel heterotachy:

```
% bali-phy 5S-rRNA/25-muscle.fasta --test -S 'GTR +> ASRV.Gamma +>
  Covarion.Huelsenbeck02'
```

You can also use the Huelsenbeck model with the free-rates model:

```
% bali-phy 5S-rRNA/25-muscle.fasta --test -S 'GTR +> ASRV.Free +>
  Covarion.Huelsenbeck02'
```

10.4. Fixing parameter values

Return to the examples directory:

```
% cd ~/alignment_files/examples/
```

We can use the TN93+Gamma(4)+Inv model without specifying parameters:

```
% bali-phy Mx/Mx_aln.nonstop.fasta --test -S 'TN93 +> ASRV.Gamma +> Inv'
```

However, we can also fix parameter values:

```
% bali-phy Mx/Mx_aln.nonstop.fasta --test -S 'TN93 +> ASRV.Gamma(n=4,alpha=1) +>
  Inv(pInv=0.2)'
```

Here we have set the shape parameter for the Gamma distribution to 1, and the fraction of invariant sites to 20%. Since these parameters are fixed, they will not be estimated and their values will not be shown in the log file.

You can see the parameters for a model by using the `help` command, as in:

```
% bali-phy help ASRV.Gamma
```

This will show the default value or default prior for each parameter, if there is one.

10.5. Priors

By default the fraction of invariant sites follows a Uniform(0,1) distribution:

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

However, we can specify an alternative prior:

```
% bali-phy Mx/Mx_aln.nonstop.fasta --test -S 'TN93 +> ASRV.Gamma(n=4) +>
  Inv(pInv~Uniform(0,0.2))'
```

We can also specify parameters as positional arguments instead of using variable names:

```
% bali-phy Mx/Mx_aln.nonstop.fasta --test -S 'TN93 +> ASRV.Gamma(4) +>
  Inv(~Uniform(0,0.2))'
```

Here "`~`" indicates a sample from the uniform distribution instead of the distribution itself.

The insertion-deletion model also has parameters.

```
% bali-phy help RS07
```

Here the default value for RS07:meanLength is ShiftedExponential(10,1). This indicates a random value that is obtained by sampling an Exponential random variable with mean 10 and then adding 1 to it.

11. Specifying the model for each partition

For analyses with multiple partitions, we might want to use different models for different partitions. When two partitions have the same model, we might also want them to have the same parameters. This is described in more detail in the manual.

11.1. Using different substitution models

Now let's specify different substitution models for different partitions.

```
% cd ~/alignment_files/examples/ITS
% bali-phy {ITS1,5.8S,ITS2}.fasta -S 1:GTR -S 2:HKY85 -S 3:TN93 --test
```

11.2. Disabling alignment estimation for some partitions

We can also disable alignment estimation for some, but not all, partitions:

```
% bali-phy {ITS1,5.8S,ITS2}.fasta -I 1:RS07 -I 2:none -I 3:RS07 --test
```

Specifying `-I 2:none` removes the insertion-deletion model and parameters for partition 2 and also disables alignment estimation for that partition.

Note that there is no longer an I3 indel model. Partition #3 now has the I2 indel model.

11.3. Sharing model parameters between partitions

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

```
% bali-phy {ITS1,5.8S,ITS2}.fasta -S 1,3:GTR -I 1,3:RS07 -S 2:TN93 -I 2:none --test
```

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

Take a look at the model parameters, and the parentheticals after the model descriptions. You should see that there is no longer an S3 substitution model or an I3 indel model. Instead, partitions #1 and #3 share the S1 substitution model and the I1 indel model.

11.4. Sharing substitution rates between partitions

We can also specify that some partitions share the same scaling factor for branch lengths:

```
% bali-phy {ITS1,5.8S,ITS2}.fasta -S 1,3:GTR -I 1,3:RS07 -S 2:TN93 -I 2:none --
  scale=1,3: --test
```

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

Take a look at the model parameters. There is no longer a scale3 parameter. Instead, partitions 1 and 3 share scale1.

12. Dataset preparation

12.1. Splitting and merging Alignments

You might want to split concatenated gene regions:

```
% cd ~/alignment_files/examples/ITS-many/
% alignment-cat -c1-223 ITS-region.fasta > 1.fasta
% alignment-cat -c224-379 ITS-region.fasta > 2.fasta
% alignment-cat -c380-551 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
```

12.2. Shrinking the data set

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

```
% alignment-thin --cutoff=5 ITS-region.fasta --verbose --protect=Csaxicola420 > ITS-
  region-thinned.fasta
```

This removes sequences with fewer than 5 directional mismatches to the closest other sequence.

- The `--verbose` option shows which sequences are removed and the distance to their closest neighbor.
- The `--protect` option keeps specific sequences in the data set.

You can also specify that data sets should be shrunk to a specific number of sequences:

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

To see all options, run:

```
% alignment-thin --help
```

12.3. Cleaning the data set

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-gappy=10 ITS-region.fasta > ITS-region-sane.fasta
```

Keep only columns with a minimum number of residues:

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