Tutorial for bppsuite

Laurent Guéguen
(laurent.gueguen@univ-lyon1.fr)

October 4, 2024

In this tutorial, we will see how simulation of sequence evolution can be performed using bio++ libraries and `bppseqgen` program, on which all information is available there:
`https://github.com/BioPP/bpp-documentation/wiki`

All the options are described in bppsuite manual:
`https://pbil.univ-lyon1.fr/bpp-doc/bppsuite/bppsuite.html`
We will specifically explore the simulation of protein algnments, but those approaches are also available for nucleotide and codon alignments.

A simulation follows a subtitution process, which is defined by a (set of) substitution models, a tree, and in option a root distribution and a substitution rate distribution.

- Many amino acid models are available in bio++, see `https://pbil.univ-lyon1.fr/bpp-doc/bppsuite/bppsuite.html#Protein`, and we will more and more complex ways to perform simulartions followign sich models.
- In all simulations, a tree is needed to guide the evolution process. First, we consider that the tree is in file `../../Data/LSUrooted.dnd`.

In the next sections, we will show more and more complex ways of simulation, using more and more information from given data.

1 Homogeneous in time and space

First, we describe examples where the modeling defined the same on all branches and sites.

1.1 Homogeneous simple model

In `simple_hom.bpp`, we consider the simplest case of evolution: homogeneous, stationnary with a simple model WAG01[1], following a tree in file `LSU.dnd`. Both are gathered in process `process1`. `simul1` sets the simulation, using this process, of a 300 amino acids protein alignment.

```
bppseqgen param=simple_hom.bpp
```

In `simul1`, the argument `output.internal.sequences = true` means that simulated ancestral sequences are output, otherwise there are only leaves sequences. The sequences are labeled with the number of the node in the tree.

[1] `https://pbil.univ-lyon1.fr/bpp-doc/bpp-phyl/html/classbpp_1_1WAG01.html#details`

1.2 Homogeneous mixture model

Instead of using a same model for all sites, it has been shown more accurate to mixtures of models, defining either amino acid properties or profiles. In the next examples, we use the models LGL08_CAT[2] that use a given number of amino acids profiles as found in databases.

In addition, we introduce a macro (NBCAT) that can be set from the command line:

```
bppseqgen param=mixed_hom.bpp NBCAT=40
```

In this file, two simulations are performed.

There are many ways to organize the mixing of models in a tree, depending how the choice of a model on a branch depends on the choice on the upper branch. In file `mixed_hom.bpp`, the line

```
scenario1= split(model=1)
```

defines a simple **scenario** of organization of the models: a site will follow the same model all along the tree, which with CAT means the same profile. `process1` follows this scenario, for `simul1`.

Without this option, at each node the process can switch freely between submodels of the mixture. `process2` defines such a process, and is used in simulation `simul2`.

Much more elaborate scenarios can be defined with the definition of **paths** (see below?)

1.3 Substitution rates

In the previous examples, all sites evolve at the same rate, whereas we could wish an heterogeneity in these rates. This can be done in two ways, either on a globally or site specifically (see section 2).

In this example, we define a distribution of substitution rates:

```
rate_distribution1 = Invariant(p=0.15,dist=Gamma(n=4))
```

which is here a discrete Gamma distribution with 4 classes and a probability 0.15 of constant sites (null rate).

1.4 Root frequencies

In the previous descriptions, the modeling is stationary, which means that the states distribution at the root is the stationary distribution of the model.

It is possible to set specific root frequencies (or states, see below) to define non-stationary process. In file `mixed_IG_hom.bpp`, a `root_freq` is defined, and used in the definition of `process2` as the root frequencies, which will be used for `simul2`.

Both simulations can be run with command:

```
bppseqgen param=mixed_IG_hom.bpp NBCAT=40
```

2 Non-homogeneous in space

In the previous modelings, the process is defined similarly for all sites (even though with mixture models a site-specific choice of submodels is done). We give how to define site-specificity, on the root states, on the substitution rates, then on the models.

This is independent of modeling time non-homogeneity, so both can be used together.

[2] `https://pbil.univ-lyon1.fr/bpp-doc/bpp-phyl/html/classbpp_1_1LGL08__CAT.html#details`.

2.1 Site-specific root states

States at the root of the tree can be set or sampled. In the tsv file `infos_rates.tsv`, column `States` assigns states to sites at root. In `sites_rate_state_hom.bpp`, this information is used for `simul1`. The column of states can be named differently, as long as it fits with `input.infos.states`.

Another way to define states at the root is to use an alignment from a given file. Still in `sites_rate_state_hom.bpp`, `input.data1` is an alignment read in file `LSU.phy`, from which a random sequence is used as a root in `simul2`.

If the simul uses a process with defined `root_freq`, the states defined with `input.infos.states` are prioritary.

2.2 Site-specific substitution rates

Rates can also set up in a site-specific way, in a similar tsv file. For example in column `Rates` of file `infos_rates.tsv`, middle sites are slower than extreme sites, as used in `simul3` of `sites_rate_state_hom.bpp`.

If the simul uses a process with defined `rate`, the rates defined with `input.infos.rates` are prioritary.

The rates can be defined without the root states (in which case those states are sampled from a distribution). In `simul4`, they are taken from a sequence in `input.data1`. In this case, only the first 981 sites are considered (to fit with the number of rates in `infos_rates.tsv`).

```
bppseqgen param=sites_state_rate_hom.bpp NBCAT=40
```

still with the site-specific rates defined in `infos_rates.tsv`.

2.3 Site-specific processes

Beyond site-specific rates and root states, it is also possible to set up site-specific processes. In file `partition.bpp`, two simple processes are defined, following the same stationary WAG01 model but different trees. Then they are assigned to different sets of sites, in a new process, which is used for simulation.

```
bppseqgen param=partition.bpp
```

Instead of preset boundaries in the partition, processes can be put into a markovian chain along the sequence, or more simply with autocorrelation probabilities of successive presence, as in `autocorr.bpp` for the same processes as before.

3 Non-homogeneous in time

In addition to the non-homogeneity in space, it is possible, totally independently, to model non-homogeneity in time, which means that different models can be assigned to different branches of the tree.

3.1 Assigned non-homogeneity

Still in file `nonstat_WAG_nonhom.bpp`, `process1` in a non-homogeneous process made of two WAG01+F[1] with different equilibrium frequencies, assgined to two sets of branches, and specific root frequencies.

`simul1` performs simple simulations with this model, whereas `simul2` in addition uses site-specific rates

3.2 Non-homogeneous mixtures

This can be done also with mixture models, and all possibilities of usage of submodels can be set within scenarios. In `nonstat_CAT_nonhom.bpp`, two CAT models are used, and with the description:

```
scenario1= split(model=(1,2))
```

all paths where a same submodel of each CAT model is used on all branches where this model is set. For example, with command:

```
bppseqgen param=nonstat_CAT_nonhom.bpp NBCAT=10
```

all 100 possible paths are with 2 submodels of CAT 10 are possible.

4 Posterior simulations

In all previous examples, simulations were done given models, root frequencies, branch lengths. But in the prospect of simulations as real as possible, it is difficult to choose without information about real data. In bio++, all those parameters can be estimated through maximum likelihood inference (via `bppml`), and given as inputs for simulations. But because of the numerous hypotheses set in the modeling (such as site-independence, too simple models, etc), the resulting simulations can still be very unrealistic.

Another way to include data information in the simulation is to use posterior transition probabilities. On a branch and a site, if the used model is M and the data D, for two states x and y, instead of using $P(y|x, M)$ as transition probabilities, we use $P(y|x, M, D)$.

Hence site and branch specific information is taken into account, even if it is not explicitly modeled.

In bio++, the `phylo` object links a `process` with `data`, providing a likelihood. To simulate alignments a posteriori, just switch in `simul` description the reference to `process` with a reference the `phylo`.

In a complete posterior simulation, the sequences simulated at the leaves will be exactly those of the given alignment, which is not very interesting. So it is possible to release this to use directly the models (and the prior probabilities) on some branches. In `simul`, argument `nullnodes` accepts a list of branch numbers, where the data will not be taken into account.

In the `simple_data.bpp` file, `nullnodes` is set to shortcut `Leaves`, which means that the simulation on all external branches is following the model.

```
bppseqgen param=simple_data.bpp
```

Beware that such simulations takes more time than previouly, since all the conditional likelihoods have to be computed.