beautier: BEAUti for R

Richèl J.C. Bilderbeek, Rampal S. Etienne

**1.** Here, we present a package, beautier, ’BEAUti for R’, for the R programming language.  
**2.** beautier allows for scripted use of the BEAST2 phylogenetics tool, by creating BEAST2 input files from an R function call.  
**3.** We describe beautier usage, the novel functionality it provides compared to BEAUti, and give some minimal examples.  
**4.** As beautier is free, libre, open-source and designed to be extended, We conclude by describing the current development of the package

# Introduction

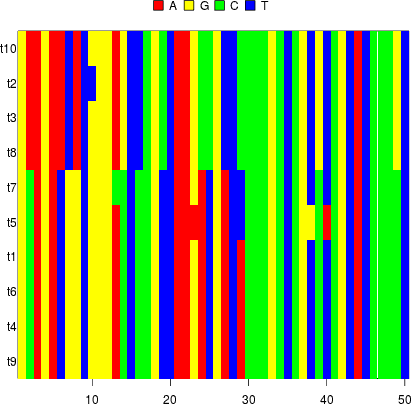
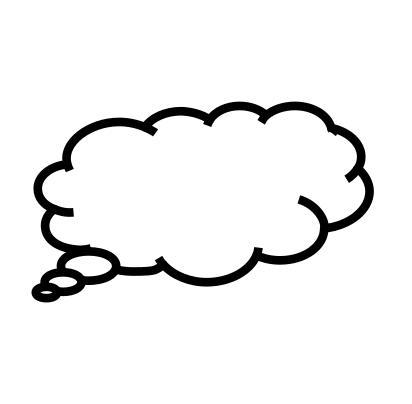
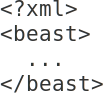
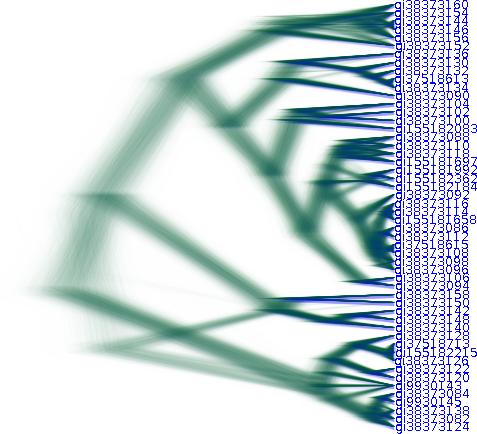
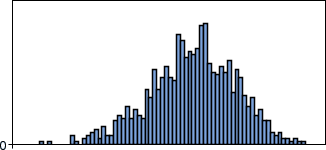
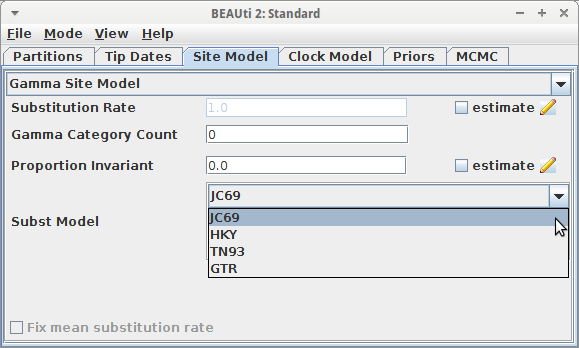
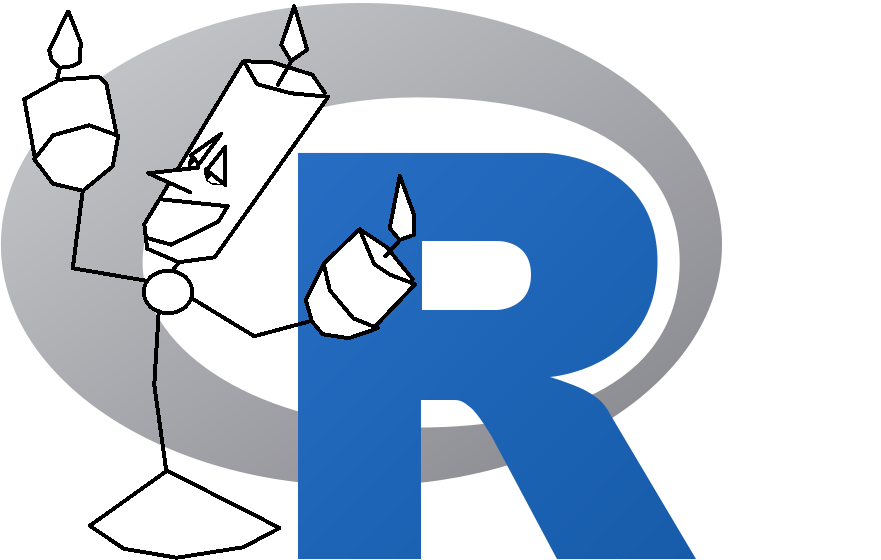
Phylogenies are a commonly used tool to explore evolutionary hypotheses. Not only can phylogenies tell us how species relate to each other, also relevant parameters like extinction and speciation rates can be estimated from them.

BEAST2 is a Bayesian phylogenetics tool. BEAST2 creates a posterior of jointly-estimated phylonies and model parameters, from a DNA, RNA or amino acid alignment. BEAST2 is a console application, that needs a configuration file containing alignments and model parameters.

BEAST2 is bundled with the BEAUti . BEAUti is a program to create a BEAST2 XML configuration file, using a user-friendly graphical user interface, with helpful and reasonable default settings. BEAUti replaces the manual editing of the BEAST2 XML files.

BEAUti cannot be called from a command-line script, which is not a problem in all cases. For example, the BEAST book encourages to first infer a posterior from simpler models first, then exploring if adding complexity changes the inferred results. This can easily be done manually using BEAUti. For bigger theoretical explorations (for example, using thousands of simulated alignments), this approach becomes inviable.

Here, we present beautier, ’BEAUti for R’, which creates BEAST2 configuration files from an R function call. The interface of beautier mimics BEAUti and is easy to use. This familiar interface helps both beginner and experienced BEAST2 users to create configuration files from shell scripts.

=[] (A) [rectangle]   ; (E) [below of=A, rectangle]  ; (F) [rectangle, below of=E]   ; (A) edge [anchor = east] node  (E) (A) edge [anchor = west] node  (E) (E) edge [anchor = east] node  (F) (E) edge [anchor = west] node  (F);

# Description

beautier is written in the R programming language . beautier creates the BEAST2 input files from an R function call, in a similar way that BEAUti does.

beautier’s main function is create\_beast2\_input\_file, which creates an BEAST2 input file. create\_beast2\_input\_file needs at least the name of a FASTA file containing a DNA alignment and a name for the to-be-created output file. This interface follows BEAUti’s default settings. Per alignment, a site model, clock model and tree priors can be chosen. Multiple alignments can be used, each with its own (unlinked) site model, clock model and tree prior.

beautier’s functions

|  |  |
| --- | --- |
| **Name** | **Description** |
| create\_beast2\_input\_file | Creates a BEAST2 input file |
| create\_gtr\_site\_model | Create a GTR site model |
| create\_hky\_site\_model | Create an HKY site model |
| create\_jc69\_site\_model | Create a Jukes-Cantor site model |
| create\_tn93\_site\_model | Create a TN93 site model |
| create\_rln\_clock\_model | Create a relaxed log-normal clock model |
| create\_strict\_clock\_model | Create a strict clock model |
| create\_bd\_tree\_prior | Create a birth-death tree prior |
| create\_cbs\_tree\_prior | Create a coalescent Bayesian skyline tree prior |
| create\_ccp\_tree\_prior | Create a coalescent constant-population tree prior |
| create\_cep\_tree\_prior | Create a coalescent exponential-population tree prior |
| create\_yule\_tree\_prior | Create a Yule tree prior |
| create\_beta\_distr | Create a beta distribution |
| create\_exp\_distr | Create an exponential distribution |
| create\_gamma\_distr | Create a gamma distribution |
| create\_inv\_gamma\_distr | Create an inverse gamma distribution |
| create\_laplace\_distr | Create a Laplace distribution |
| create\_log\_normal\_distr | Create a log-normal distribution |
| create\_normal\_distr | Create a normal distribution |
| create\_one\_div\_x\_distr | Create a 1/X distribution |
| create\_poisson\_distr | Create a Poisson distribution |
| create\_uniform\_distr | Create a uniform distribution |

In total, beautier has 59 exported functions to create a BEAST2 configuration file. beautier is an alternative for a majority of BEAUti use cases. beautier does not support the full functionality of BEAUti. Considering BEAUti’s flexibility and number of plugins, this would be a Herculean effort. To compensate for this, an extensible software architecture is used. beautiers future extensions can be found on its GitHub.

BEAUti assumes that a phylogeny has a crown age that needs to be jointly-estimated with the phylogeny and other parameters. BEAUti does not allow for fixing a phylogeny’s crown age. Before beautier, one needs to manually edit the BEAST2 XML configuration file, which is prone to errors. beautier, allows easy fixing of phylogenies’ crown ages. For theoretical work, a fixed and known crown age can result in a cleaner analysis.

beastier and lumier are related packages, used in testing. lumier calls BEAST2 from within R. lumier is used to confirm that the XML files created by beautier are valid. Additionally, lumier is used to run BEAST2 to create posteriors. Using beastier, these posteriors are checked to have an estimated or fixed crown age.

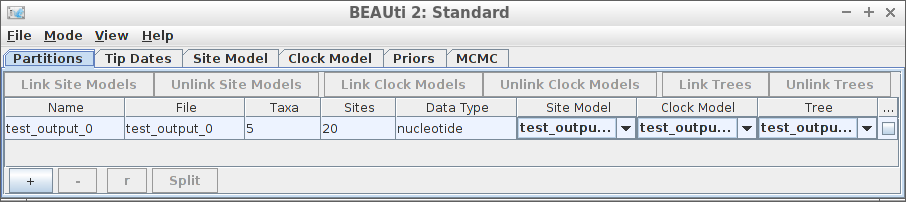
# Examples

In R, a package’s function need to be loaded in the global namespace first:

library(beautier)

BEAUti, and likewise beautier, need at least a FASTA filename and an XML output filename. In BEAUti, this is achieved by loading a FASTA file (resulting in figure [fig:simplest\_beauti\_usage]), then saving an output file using a common save file dialog. In beautier, the same is achieved by listing [lst:simplest\_example]:

library(beautier)  
create\_beast2\_input\_file(  
 "alignment.fas",  
 "beast2.xml"  
)



Simplest BEAUti usage

This code will create a BEAST2 file with name ’beast2.xml’, using a FASTA file with name alignment.fas, using the same default settings as BEAUti. The default settings are, among others, to use a Jukes-Cantor site model , a strict clock, and a Yule birth tree prior .

An example of using a different site model, clock model and tree prior is shown by listing [lst:all\_different]:

library(beautier)  
create\_beast2\_input\_file(  
 "alignment.fas",  
 "beast2.xml",  
 site\_models = create\_hky\_site\_model(),  
 clock\_models = create\_rln\_clock\_model(),  
 tree\_priors = create\_bd\_tree\_prior()  
)

This code uses an HKY site model, a relaxed log-normal clock model and a birth-death tree prior . Table [tab:functions] shows an overview of all functions to create site models, clock models and tree priors.

The argument names site\_models, clock\_models and tree\_priors are plural, as each of these can be (a list of) one or more elements. Each of these arguments must have the same number of elements, so that each alignment has its own site model, clock model and tree prior.

beautier creates site models, clock models and tree priors with the same default distributions as BEAUti. For example, a Yule tree prior assumes that birth rate likelihoods follow a uniform distribution, from minus infinity to infinity. This assumption entails that negative and positive birth rates are just as likely, where a negative birth rate is biologically impossible. One may prefer to have an exponential distribution instead, as this would state that birth rates are always positive, and higher values are less likely than lower values. To do so beautier is shown by listing [lst:diff\_distr]:

library(beautier)  
create\_beast2\_input\_file(  
 "alignment.fas",  
 "beast2.xml",  
 tree\_priors = create\_yule\_tree\_prior(  
 birth\_rate\_distr = create\_exp\_distr()   
 )  
)

Novel about beautier is that it allows for specifying a fixed crown age. By default, a phylogeny’s crown age is jointly-estimated with the other parameters. Setting a fixed crown age is not yet possible in BEAUti directly, but it is documented how to manually edit the XML file to allow for a fixed crown age. Listing [lst:fixed\_crown\_age] shows how to specify a fixed crown age with beautier:

create\_beast2\_input\_file(  
 "alignment.fas",  
 "beast2.xml"  
 posterior\_crown\_age = 15  
)

# beautier development and other resources

beautier is free, libre and open source software available from the official R package archive at <http://cran.r-project.org/src/contrib/PACKAGES.html#beautier>. beautier is licensed under the GNU General Public License.

beautier’s development takes place on GitHub , which is a good practice for computational scientists and improves transparency .

beautier’s uses the Travis CI continuous integration service, which is known to significantly increase the the number of bugs exposed . beautier has a 100% code coverage, which correlates with code quality . beautier follows Hadley Wickham’s style guide , which improves software quality .

beautier is dependent on multiple packages, which are APE , beastier , devtools , geiger , ggplot2 , knitr , lumier , phangorn , rmarkdown , seqinr , stringr , testit and TreeSim .

beautier’s documentation is extensive, yet concise. All functions are documented in the package’s internal documentation. For quick use, each exported function shows a minimal example. For easy exploration, each exported function’s documentation links to related functions. Additionally, beautier has a vignette that demonstrates in a longer form how to use it. The integrity of this documentation is tested by Travis CI. The GitHub documentation helps to get started, with a dozen examples of a BEAUti screenshot and the equivalent beautier code.

beautier’s GitHub facilitates feature requests and has guidelines how to do so. Thanks to Travis CI, newly submitted code is expected to be accepted quicker .

# Citation of beautier

Scientists using beautier in a published paper should cite this article. Users can additionally cite the beautier package directly. Citation information can be obtained by typing:

> citation("beautier")

from within R.