beautier: BEAUti for R

Richèl J.C. Bilderbeek, Rampal S. Etienne

**Summary**

**1.** In the field of phylogenetics, BEAST2 is one of the most widely used software tools. It comes with the graphical program BEAUti to facilitate the creation of input files to BEAST2. However, when many input files are needed, such a GUI is cumbersome. Moreover, many other phygenetics tools are available in R, which requires switching from one platform to the other.

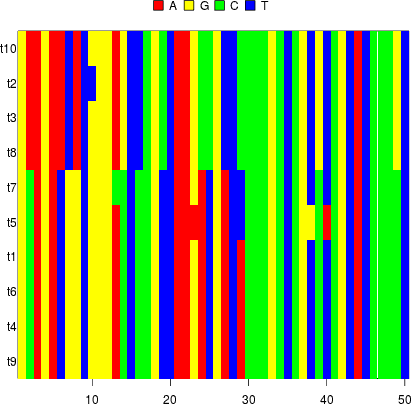
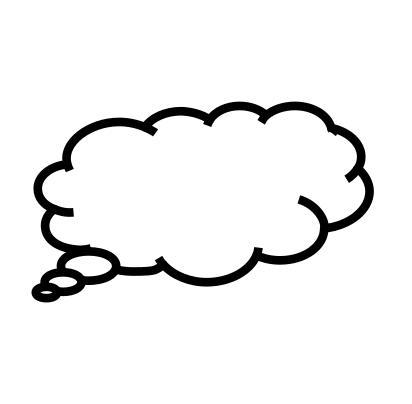
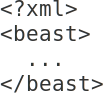
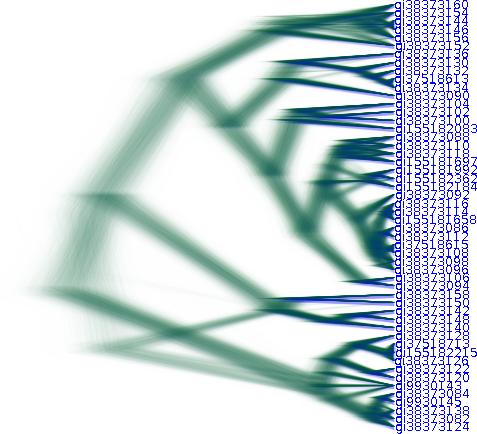
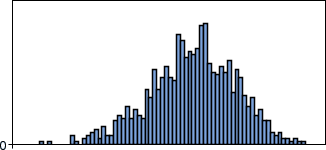
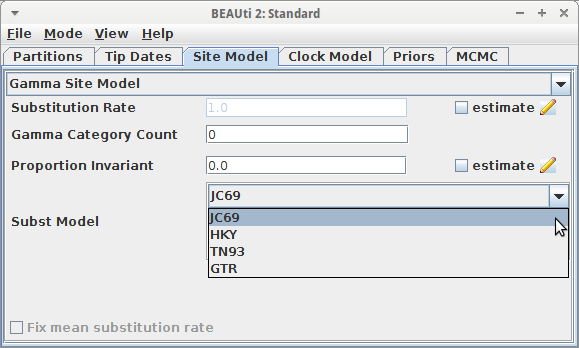
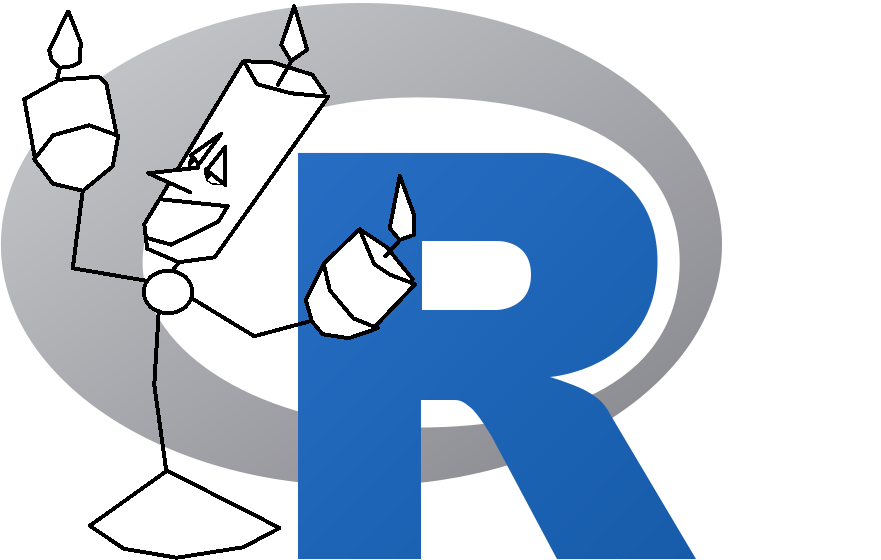
2, Here, we present a free, libre and open-source package, beautier, ’BEAUti for R’, for the R programming language.beautier creates BEAST2 input files from an R function call.**3.** We describe beautier’s usage, the novel functionality it provides compared to BEAUti, and give some examples.  
**4.** As beautier is designed to be of high quality and extendable, we conclude by describing the further development of the package

# Introduction

Phylogenies are commonly used to explore evolutionary hypotheses. Not only can phylogenies show us how species (or other evolutionary units) relate to each other, but also relevant parameters such as extinction and speciation rates can be estimated from them.There are many phylogenetics tools available to obtain an estimate of the phylogenetic tree of a given set of species. BEAST2 is one of the most widely used ones. It creates a posterior of jointly-estimated phylogenies and model parameters, from a DNA, RNA or amino acid alignment (see figure [fig:workflow] for an overview of the workflow). It is a console application, that needs a configuration file containing alignments and model parameters.BEAST2 is bundled with BEAUti , a desktop application to create a BEAST2 configuration file. BEAUti has a user-friendly graphical user interface, with helpful and reasonable default settings. As such, BEAUti is an attractive alternative to manual and error-prone editing of BEAST2 configuration files.BEAUti cannot be called from a command-line script. This implies that when the user wants to explore the consequences of various settings, this must be done manually.This is the common workflow when using a few alignments and doing a superficial analysis of sensitivity of the reconstructed tree to model settings. However, for exploring many trees (for instance from simulations) and formore thorough sensitivity analysis, , one would like to loop through multiple (simulated) alignments, nucleotide substitution models, clock models and tree priors.

Here, to provide such functionality we present beautier, ’BEAUti for R’, which creates BEAST2 configuration files from an R function call. and hence will save time and tedious mouse clicking, and reduces the chances of errors in such repetitive actions. The interface of beautier mimics BEAUti. This familiarity helps both beginner and experienced BEAST2 users to create configuration files using beautier. Because there are many R packages for exploring the resulting phylogenetic trees, beautier enables the creation of a single-script pipeline from sequence alignments to tree analysis in R.

beautier has a similar goal as BEASTmasteR , as both allow for scripted use of BEAST2. The difference is that beautier has its focus on ultrametric trees, where BEASTmasteR is used for tip-dating.

=[] (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 and creates a BEAST2 configuration file from an R function call, in a similar way that BEAUti does.beautier’s main function is create\_beast2\_input\_file, which, as the name suggests, creates a BEAST2 configuration 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 configuration file. The default settings for the other arguments of create\_beast2\_input\_file are identical to BEAUti’s default settings. Per alignment, a site model, clock model and tree prior can be chosen. Multiple alignments can be used, each with its own (unlinked) site model, clock model and tree prior.

beautier’s main 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 |

beautier currently has 61 exported functions to create a BEAST2 configuration file. beautier is an alternative for a majority of BEAUti use cases, but does not yet support the full functionality of BEAUti. Because of BEAUti’s high number of plugins, beautier uses an extensible software architecture. A roadmap of beautier’s future extensions can be found on its GitHub.

Our initial motivation to create beautier is that we wanted to fix the crown age of a phylogeny, but because BEAUti assumes that a phylogeny has a crown age that needs to be jointly-estimated with the phylogeny and other parametersit does not allow for fixing the crown age. Without beautier, one needs to manually edit the BEAST2 XML configuration file, which is tedious and prone to errors. beautier, allows for easy setting of a fixed crown age, enabling theoretical experiments with one less source of variation.

beastier and lumier are related packages, used to confirm the correct working of beautier;. lumier calls BEAST2 from within R and is used to confirm that the XML files created by beautier are valid. Additionally, lumier can be used to run BEAST2 to create posteriors. beastier can parse BEAST2 posteriors, and can, for example, be used to confirm whether posteriors have an estimated or fixed crown age.

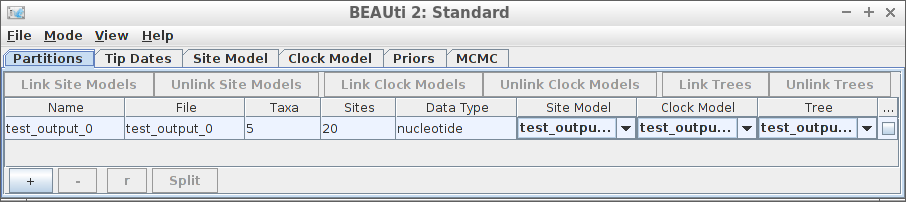
# Usage

In R, the functions of a package need to be loaded in the global namespace first:

library(beautier)

BEAUti, and likewise beautier, needs at least a FASTA filename and an 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::

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



Simplest BEAUti usage

This code will create a BEAST2 configuration 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:

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, each with their default settings in BEAUti . Table [tab:functions] shows an overview of all functions to create site models, clock models and tree priors.Note that the arguments’ 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 also uses the same default distributions as BEAUti for the site models, clock models and tree priors. For example, a Yule tree prior assumes that the birth rate follows a uniform distribution, from minus infinity to plus infinity. This assumption implies that negative and positive birth rates are just as likely, where a negative birth rate is biologically impossible[[1]](#footnote-1). One may prefer an exponential distribution instead, as this would assume only positive birth rates, and make high birth rates unlikely.

The following script shows how to do this in beautier:

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

Novel functionality that beautier offers, 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 de BEAUti documentation mentions how to manually edit the XML file to allow for a fixed crown age. This is 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](http://cran.r-project.org/src/contrib/PACKAGES.html" \l "beautier) and is licensed under the GNU General Public License v3.0. beautier’s uses the Travis CI continuous integration service, which is known to significantly increase the number of bugs exposed and increases the speed at which new features are added. beautier has a 100% code coverage, which correlates with code quality, and it follows Hadley Wickham’s style guide to improve software quality . beautier depends on multiple packages, which are APE , beastier , devtools , geiger , ggplot2 , knitr , lumier , phangorn , rmarkdown , seqinr , stringr , testit and TreeSim .

beautier’s development takes place on GitHub , which facilitates collaboration and improves transparency. beautier’s GitHub accomodates feature requests and has a guidelines how to do so.

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 extensively how to use it. The GitHub documentation helps to get started, with a dozen examples of BEAUti screenshots with equivalent beautier code.

# Citation of beautier

Scientists using beautier in a published paper can cite this article, and/or cite the beautier package directly. To obtain this citation from within an R script, use:

> citation("beautier")

1. note that in practice, this usually works out just fine [↑](#footnote-ref-1)