beautier: BEAUti for R

Richèl J.C. Bilderbeek, Rampal S. Etienne

January 16, 2018

Abstract

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

1 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 [6] 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 [12]. 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 [10] 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.

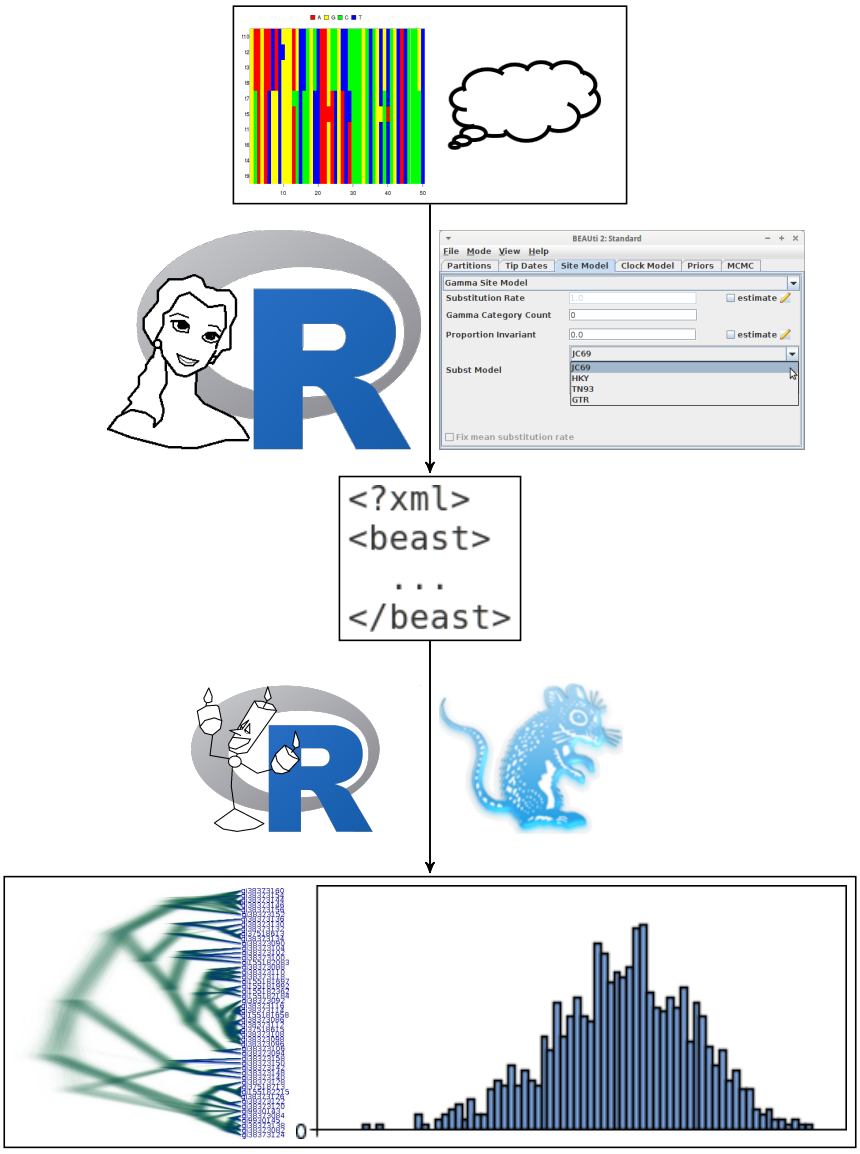

Figure 1: Workflow. From an alignment and priors, one creates a BEAST2 XML input file. This can be done using beautier or BEAUti. The created configuration file is run by lumier or BEAST2 to create a posterior of phylogenies and model parameter estimates.

Name	Description
create_beast2_input_file	Creates a BEAST2 input file
create_gtr_site_model	Create a GTR site model [26]
create_hky_site_model	Create an HKY site model [16]
create_jc69_site_model	Create a Jukes-Cantor site model [7]
create_tn93_site_model	Create a TN93 site model [25]
create_rln_clock_model	Create a relaxed log-normal clock model [11]
create_strict_clock_model	Create a strict clock model [22]
create_bd_tree_prior	Create a birth-death tree prior [18]
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 [34]
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

Table 1: beautier's functions

2 Description

`beautier` is written in the R programming language [21]. `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.

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

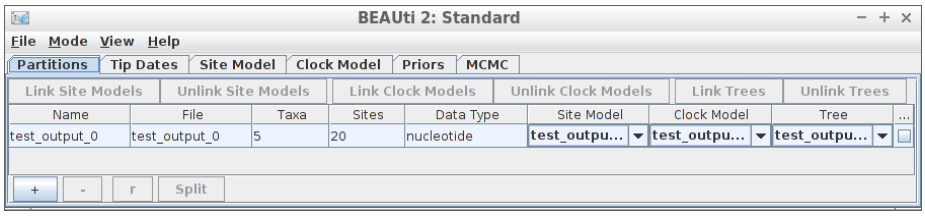

Figure 2: Simplest BEAUti usage

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.

3 Examples

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

```
library(beautier)
```

Listing 1: Loading

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 2), then saving an output file using a common save file dialog. In `beautier`, the same is achieved by listing 2:

```
library(beautier)
create_beast2_input_file(
  "alignment.fas",
  "beast2.xml"
)
```

Listing 2: Simplest example

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 [7], a strict clock, and a Yule birth tree prior [34].

An example of using a different site model, clock model and tree prior is shown by listing 3:

```
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()
)
```

Listing 3: Example with different site model and clock model and tree prior

This code uses an HKY site model, a relaxed log-normal clock model and a birth-death tree prior [18]. Table 1 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 4:

```
library(beautier)
create_beast2_input_file(
  "alignment.fas",
  "beast2.xml",
  tree_priors = create_yule_tree_prior(
    birth_rate_distr = create_exp_distr()
  )
)
```

Listing 4: Example with Yule tree prior with different birth rate distribution

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 5 shows how to specify a fixed crown age with `beautier`:

```
create_beast2_input_file(
  "alignment.fas",
  "beast2.xml"
  posterior_crown_age = 15
)
```

Listing 5: Example with fixed crown age

4 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 [1], which is a good practice for computational scientists [20] and improves transparency [14].

`beautier`'s uses the Travis CI [2] continuous integration service, which is known to significantly increase the the number of bugs exposed [27]. `beautier` has a 100% code coverage, which correlates with code quality [17, 9]. `beautier` follows Hadley Wickham's style guide [28], which improves software quality [13].

`beautier` is dependent on multiple packages, which are `APE` [19], `beastier` [4], `devtools` [31], `geiger` [15], `ggplot2` [29], `knitr` [33], `lumier` [5], `phangorn` [23], `rmarkdown` [3], `seqinr` [8], `stringr` [30], `testit` [32] and `TreeSim` [24].

`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 [27].

5 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.

References

[1] Github. `https://github.com/`.

[2] Travis CI. `https://travis-ci.org/`.

[3] JJ Allaire, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, and Winston Chang. *rmarkdown: Dynamic Documents for R*, 2017. R package version 1.8.

[4] Richel J.C. Bilderbeek. *beastier: Work with BEAST and BEAST2 output*. R package version 0.1.0.

[5] Richel J.C. Bilderbeek. *lumier: run BEAST2 from within R*. R package version 0.1.0.

[6] Remco Bouckaert, Joseph Heled, Denise Kühnert, Tim Vaughan, Chieh-Hsi Wu, Dong Xie, Marc A Suchard, Andrew Rambaut, and Alexei J Drummond. Beast 2: a software platform for bayesian evolutionary analysis. *PLoS Comput Biol*, 10(4):e1003537, 2014.

[7] Jukes Cantor and T Jukes. Mammalian protein metabolism. *Evolution of protein molecules. Academic Press, New York, NY*, pages 21–132, 1969.

[8] D. Charif and J.R. Lobry. SeqinR 1.0-2: a contributed package to the R project for statistical computing devoted to biological sequences retrieval and analysis. In U. Bastolla, M. Porto, H.E. Roman, and M. Vendruscolo, editors, *Structural approaches to sequence evolution: Molecules, networks, populations*, Biological and Medical Physics, Biomedical Engineering, pages 207–232. Springer Verlag, New York, 2007. ISBN : 978-3-540-35305-8.

[9] Fabio Del Frate, Praerit Garg, Aditya P Mathur, and Alberto Pasquini. On the correlation between code coverage and software reliability. In *Software Reliability Engineering, 1995. Proceedings., Sixth International Symposium on*, pages 124–132. IEEE, 1995.

[10] Alexei J Drummond and Remco R Bouckaert. *Bayesian evolutionary analysis with BEAST*. Cambridge University Press, 2015.

[11] Alexei J Drummond, Simon YW Ho, Matthew J Phillips, and Andrew Rambaut. Relaxed phylogenetics and dating with confidence. *PLoS biology*, 4(5):e88, 2006.

[12] Alexei J Drummond, Marc A Suchard, Dong Xie, and Andrew Rambaut. Bayesian phylogenetics with beauti and the beast 1.7. *Molecular biology and evolution*, 29(8):1969–1973, 2012.

[13] Xuefen Fang. Using a coding standard to improve program quality. In *Quality Software, 2001. Proceedings. Second Asia-Pacific Conference on*, pages 73–78. IEEE, 2001.

[14] Krzysztof J Gorgolewski and Russell Poldrack. A practical guide for improving transparency and reproducibility in neuroimaging research. *bioRxiv*, page 039354, 2016.

[15] LJ Harmon, JT Weir, CD Brock, RE Glor, and W Challenger. Geiger: investigating evolutionary radiations. *Bioinformatics*, 24:129–131, 2008.

[16] Masami Hasegawa, Hirohisa Kishino, and Taka-aki Yano. Dating of the human-ape splitting by a molecular clock of mitochondrial dna. *Journal of molecular evolution*, 22(2):160–174, 1985.

[17] Joseph R. Horgan, Saul London, and Michael R Lyu. Achieving software quality with testing coverage measures. *Computer*, 27(9):60–69, 1994.

[18] David G Kendall. On the generalized" birth-and-death" process. *The annals of mathematical statistics*, pages 1–15, 1948.

[19] E. Paradis, J. Claude, and K. Strimmer. APE: analyses of phylogenetics and evolution in R language. *Bioinformatics*, 20:289–290, 2004.

[20] Yasset Perez-Riverol, Laurent Gatto, Rui Wang, Timo Sachsenberg, Julian Uszkoreit, Felipe Leprevost, Christian Fufezan, Tobias Ternent, Stephen J Eglen, Daniel SS Katz, et al. Ten simple rules for taking advantage of git and github. *bioRxiv*, page 048744, 2016.

[21] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[22] Vincent M Sarich and Allan C Wilson. Immunological time scale for hominid evolution. *Science*, 158(3805):1200–1203, 1967.

[23] K.P. Schliep. phangorn: phylogenetic analysis in r. *Bioinformatics*, 27(4):592–593, 2011.

[24] Tanja Stadler. *TreeSim: Simulating Phylogenetic Trees*, 2017. R package version 2.3.

[25] Koichiro Tamura and Masatoshi Nei. Estimation of the number of nucleotide substitutions in the control region of mitochondrial dna in humans and chimpanzees. *Molecular biology and evolution*, 10(3):512–526, 1993.

[26] Simon Tavaré. Some probabilistic and statistical problems in the analysis of dna sequences. *Lectures on mathematics in the life sciences*, 17(2):57–86, 1986.

[27] Bogdan Vasilescu, Yue Yu, Huaimin Wang, Premkumar Devanbu, and Vladimir Filkov. Quality and productivity outcomes relating to continuous integration in github. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pages 805–816. ACM, 2015.

[28] Hadley Wickham. Style guide. `http://r-pkgs.had.co.nz/style.html`. Accessed: 2017-12-20.

[29] Hadley Wickham. *ggplot2: elegant graphics for data analysis*. Springer New York, 2009.

[30] Hadley Wickham. *stringr: Simple, Consistent Wrappers for Common String Operations*, 2017. R package version 1.2.0.

[31] Hadley Wickham and Winston Chang. *devtools: Tools to Make Developing R Packages Easier*, 2016. R package version 1.12.0.9000.

[32] Yihui Xie. *testit: A Simple Package for Testing R Packages*, 2014. R package version 0.4.

[33] Yihui Xie. *knitr: A General-Purpose Package for Dynamic Report Generation in R*, 2017. R package version 1.17.

[34] G Udny Yule. A mathematical theory of evolution, based on the conclusions of dr. jc willis, frs. *Philosophical transactions of the Royal Society of London. Series B, containing papers of a biological character*, 213:21–87, 1925.

References