babette: BEAUti 2, BEAST2 and Tracer for R

Richèl J.C. Bilderbeek, Rampal S. Etienne

February 9, 2018

Summary

1. In the field of phylogenetics, BEAST2 is one of the most widely used software tools. It comes with the graphical programs BEAUti 2 to facilitate the creation of BEAST2 configuration files, and with DensiTree and Tracer, to interpret BEAST2's output files. However, when many different alignments or model setups are required, a workflow of GUI programs is cumbersome.
2. Here, we present a free, libre and open-source package, `babette`, 'BEAUti 2, BEAST2 and Tracer for R', for the R programming language. `babette` creates BEAST2 input files, runs BEAST2 and parses its results, all from an R function call.
3. We describe `babette`'s usage, the novel functionality it provides compared to the tools it is alternative of, and give some examples.
4. As `babette` is designed to be of high quality and extendable, we conclude by describing the further development of the package

Keywords: computational biology, evolution, phylogenetics, BEAST2, R

1 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 like 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 [8] 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 1 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 2 [12] ('BEAUti' from now on), 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 for more thorough sensitivity analysis, one would like to loop through multiple (simulated) alignments, nucleotide substitution models, clock models and tree priors.

BEAST2 is also associated with Tracer [22] and DensiTree [7]. Both are desktop applications to analyze the output of BEAST2, each with a user-friendly graphical user interface.

Tracer's purpose is to analyze the parameter estimates generated from a BEAST 2 run. It shows, among others, the effective sample size (ESS) and time series ('the trace', hence the name) of each variable in the MCMC run. Both ESS and trace are needed to assess the strength of the inference. DensiTree visualizes the phylogenies of a BEAST2 posterior, with many options to improve the displayal of those many phylogenies.

However, for exploring the output of many BEAST2 runs, one would like a script to collect all parameter ESSes, parameter traces and posterior phylogenies.

Here, to provide such functionality we present `babette`, BEAUti 2, BEAST2 and Tracer for R, which creates BEAST2 configuration files, runs BEAST2, and analyzes its results, all from an R function call. This will save time, tedious mouse clicking and reduces the chances of errors in such repetitive actions. The interface of `babette` mimics the tools it is inspired on. This familiarity helps both beginner and experienced BEAST2 users to make the step from those tools to `babette`. `babette` enables the creation of a single-script pipeline from sequence alignments to posterior analysis in R.

`babette` is the first R package that unifies the full workflow of working with BEAST2. An example to create a BEAST2 input file is `BEASTmasteR` [18]. Also `BEASTmasteR` allows to create BEAST2 configuration files from R. The difference is that `babette` has its focus on DNA alignments and ultrametric trees, where `BEASTmasteR` is used for morphological traits and tip-dating. Examples of R packages to parse the BEAST2 output files are `rBEAST` [14] and `RBEAST` [**?**].

2 Description

`babette` is written in the R programming language [21] and enables the full BEAST2 workflow from an R function call, in a similar way that BEAUti, DensiTree and Tracer do.

`babette`'s main function is `run_beast2`, which, configures BEAST2, runs it and parses its output. `run_beast2` needs at least the name of a FASTA file containing a DNA alignment. The default settings for the other arguments of `run_beast2` are identical to BEAUti's and BEAST2's default settings. Per

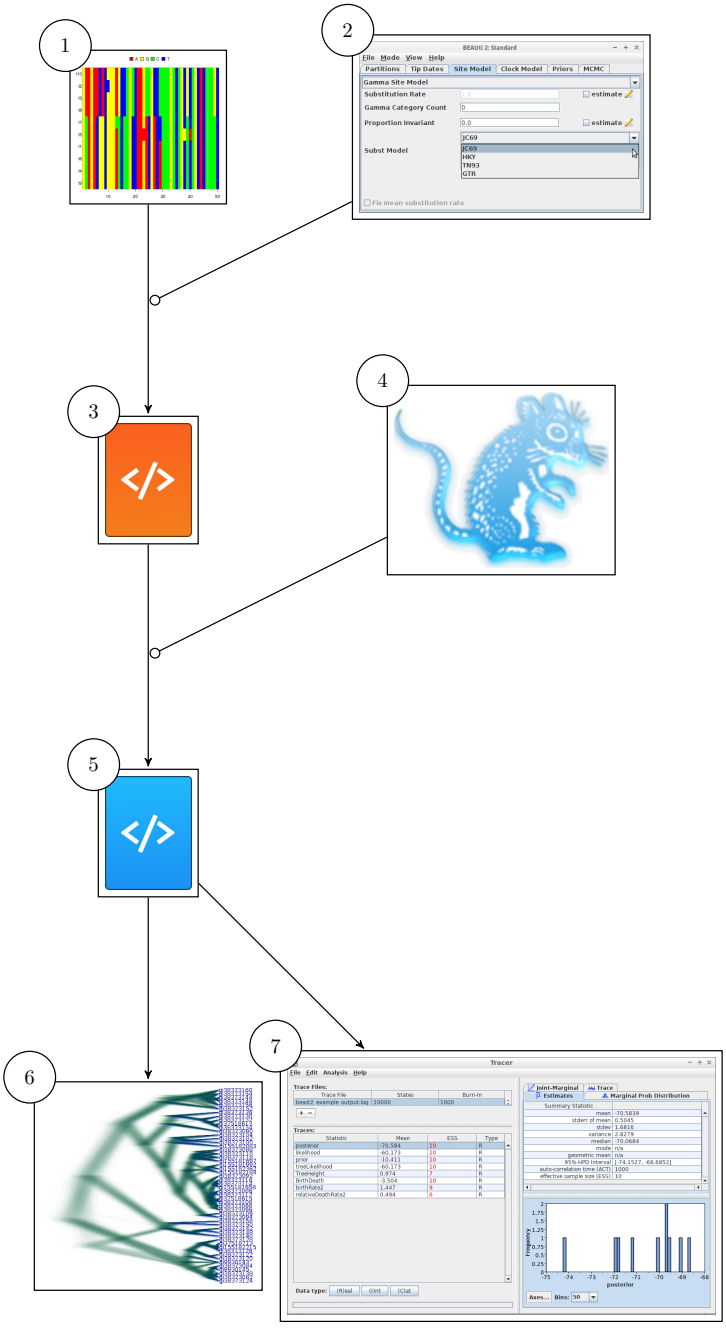

Figure 1: Workflow using GUI tools. From an alignment (1) and BEAUti (2), a BEAST2 configuration file (3) is created. BEAST2 (4) uses that file to infer a posterior, storing it in multiple files (5). These results are visualized using DensiTree (6) and Tracer (7). babette allows for the same workflow, all from an R function call.

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.

`babette` currently has 61 exported functions to set up a BEAST2 configuration file. `babette` 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, `babette` uses a software architecture that expects te be extended.

`babette` has 7 exported function to run and help run BEAST2. One function is a wrapper function to run BEAST2, others allow the user to check if a BEAST2 configuration file is indeed valid.

`babette` has 20 exported function to parse the BEAST2 output files and analyse the created posterior. `babette` gives the same ESSes and summary statistics as Tracer. The data is formatted as such, that it can easily be visualized using `ggplot2` (for a trace, similar to Tracer) or `phangorn` [23] (for the phylogenies in a posterior, similar to DensiTree).

3 Examples

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

```
library(babette)
```

BEAUti, and likewise `babette`, needs at least a FASTA filename to produce a BEAST2 configuration file. In BEAUti, this is achieved by loading a FASTA file, then saving an output file using a common save file dialog. After this, BEAST2 needs to be invoked on the created configuration file to create multiple files storing the created posterior. Finally, these output files must be parsed by either Tracer of DensiTree.

In `babette`, the same is achieved by:

```
out <- run_beast2("alignment.fas")
```

This code will create a (temporary) BEAST2 configuration file, using a FASTA file with name `alignment.fas`, using the same default settings as BEAUti. `babette` will then execute a BEAST2 with that file, and parses the output. The output, named `out`, is a list of parameter estimates, posterior phylogenies (one per alignment) and operator acceptances.

The default settings for BEAUti (and thus `babette`) are, among others, to use a Jukes-Cantor site model [9], a strict clock, and a Yule birth tree prior [32].

An example of using a different site model, clock model and tree prior is:

```
out <- run_beast2(
  "alignment.fas",
  site_models = create_hky_site_model(),
  clock_models = create_rln_clock_model(),
  tree_priors = create_bd_tree_prior()
)
```

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

Table 1: babette's main functions

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 1 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.

An example of two alignments, each with its own site model, is:

```
out <- run_beast2(
  c("anthus_aco.fas", "anthus_nd2.fas"),
  site_models = list(
    create_tn93_site_model(),
    create_gtr_site_model()
  )
)
```

`babette` also uses the same default distributions as BEAUti for the site models, clock models and tree priors. For example, a Yule tree prior assumes 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 (note that in practice, this usually works out just fine). One may prefer an exponential distribution instead, as this would assume only positive birth rates, and makes high birth rates unlikely.

The following script shows how to do this in `babette`:

```
out <- run_beast2(
  "alignment.fas",
  tree_priors = create_yule_tree_prior(
    birth_rate_distr = create_exp_distr()
  )
)
```

Our initial motivation to create `babette` is that we wanted to fix the crown age of a phylogeny. BEAUti assumes that a phylogeny has a crown age that needs to be jointly-estimated with the phylogeny and other parameters. It does not allow for fixing the crown age. Without `babette`, one needs to manually edit the BEAST2 configuration file, which is tedious and prone to errors. Fixing the crown ages is especially useful for theoretical experiments, as this allows for one less source of variation.

This is how to specify a fixed crown age with `babette`:

```
out <- run_beast2(
  "alignment.fas",
  posterior_crown_age = 15
)
```

`babette` allows for the same functionality of Tracer. Tracer, among others, shows the effective sample sizes of each posterior's variables. As an MCMC run needs to converge first, Tracer discards the first 10% of all parameter estimates. This is called the burn-in. To calculate the effective sample sizes in `babette`, with the same burn-in of 10%:

```
traces <- remove_burn_ins(traces = out$estimates, burn_in_
    fraction = 0.1)
esses <- calc_esses(traces, sample_interval = 1000)
```

Tracer displays multiple summary statistics for each estimated variable: the mean and its standard error, standard deviation, variance, median, mode, geometric mean, 95% highest posterior density interval, auto-correlation time and effective sample size. To obtain all these summary statistics of, for example, the estimated birth rate in `babette`:

```
sum_stats <- calc_sum_stats(
  out$estimates$birthRate,
  sample_interval = 1000,
  burn_in_fraction = 0.1
)
```

`babette` allows for the same functionality of DensiTree. DensiTree displays the phylogenies in a posterior at the same time scale over one another, allowing

to see the uncertainty in topology and branch lengths. To visualize the same in babette:

```
densitree(out$anthus_aco_trees)
```

4 babette development and other resources

babette is free, libre and open source software available from the official R package archive at http://cran.r-project.org/src/contrib/PACKAGES.html#babette and is licensed under the GNU General Public License v3.0.

babette uses the Travis CI [2] continuous integration service, which is known to significantly increase the the number of bugs exposed [25] and increases the speed at which new features are added [25]. babette has a 100% code coverage, which correlates with code quality [17, 11]. babette follows Hadley Wickham's style guide [26], which improves software quality [13]. babette is dependent on multiple packages, which are APE [19], beautier [5], beastier [4], devtools [29], geiger [16], ggplot2 [27], knitr [31], phangorn [23], rmarkdown [3], seqinr [10], stringr [28], testit [30] and tracerer [6] and TreeSim [24].

babette's development takes place on GitHub [1], https://github.com/richelbilderbeek/babette, which accommodates collaboration [20] and improves transparency [15]. babette's GitHub facilitates feature requests and has a guidelines how to do so.

babette's documentation is extensive. 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, babette 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 babette code.

5 Citation of babette

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

```
> citation("babette")
```

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: BEAST2 from R*. R package version 1.0, `https://github.com/richelbilderbeek/beastier`.

[5] Richel J.C. Bilderbeek. *beautier: BEAUti 2 from R*. R package version 1.0, `https://github.com/richelbilderbeek/beautier`.

[6] Richel J.C. Bilderbeek. *tracerer: Tracer from R*. R package version 1.0, `https://github.com/richelbilderbeek/tracerer`.

[7] Remco Bouckaert and Joseph Heled. Densitree 2: Seeing trees through the forest. *bioRxiv*, page 012401, 2014.

[8] 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.

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

[10] 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.

[11] 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.

[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] Nuno Faria and Marc A. Suchard. *RBeast*.

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

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

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

[18] Nicholas J. Matzke. *BEASTmasteR: R tools for automated conversion of NEXUS data to BEAST2 XML format, for fossil tip-dating and other uses.* R package version 0.2.

[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] Andrew Rambaut and Alexei J Drummond. *Tracer v1.4*, 2007. Available from http://beast.bio.ed.ac.uk/Tracer.

[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] 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.

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

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

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

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

[30] Yihui Xie. *testit: A Simple Package for Testing R Packages*, 2014. R package version 0.4, `http://CRAN.R-project.org/package=testit`.

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

[32] 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.