

1 babette: BEAUti 2, BEAST2 and Tracer for R

2 Richèl J.C. Bilderbeek¹ and Rampal S. Etienne¹

3 ¹Groningen Institute for Evolutionary Life Sciences, University of
4 Groningen, Groningen, The Netherlands

5 March 5, 2018

6 Summary

7 **1.** In the field of phylogenetics, BEAST2 is one of the most widely
8 used software tools. It comes with the graphical user interfaces BEAUti
9 2, DensiTree and Tracer, to create BEAST2 configuration files and to in-
10 terpret BEAST2's output files. However, when many different alignments
11 or model setups are required, a workflow of graphical user interfaces is
12 cumbersome.

13 **2.** Here, we present a free, libre and open-source package, **babette**:
14 'BEAUti 2, BEAST2 and Tracer for R', for the R programming language.
15 **babette** creates BEAST2 input files, runs BEAST2 and parses its results,
16 all from an R function call.

17 **3.** We describe **babette**'s usage and the novel functionality it provides
18 compared to the original tools and we give some examples.

19 **4.** As **babette** is designed to be of high quality and extendable, we
20 conclude by describing the further development of the package.

21
22 **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) are related to each other, but we also estimate relevant parameters such as extinction and speciation rates. There are many phylogenetics tools available to obtain an estimate of the phylogenetic tree of a given set of species. BEAST2 (Bouckaert *et al.* 2014) is one of the most widely used ones. It creates a posterior of jointly estimated phylogenies and model parameters, from one or more DNA, RNA or amino acid alignments (see figure 1 for an overview of the workflow). It has a graphical and a command-line interface, that both need a configuration file containing alignments and model parameters. BEAST2 is bundled with BEAUti 2 (Drummond *et al.* 2012) ('BEAUti' from now on), a desktop application to create a BEAST2 configuration file. BEAUti has a user-friendly graphical user interface, with helpful default settings. As such, BEAUti is an attractive alternative to manual and error-prone editing of BEAST2 configuration files.

However, 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 manageable workflow when using a few alignments and doing a superficial analysis of sensitivity of the reconstructed tree to model settings. 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. One such tool to replace BEAUti is **BEASTmasterR** (Matzke 2015), which focuses on morphological traits and tip-dating, but also supports DNA data. **BEASTmasterR**, however, requires hundreds of lines of R code to setup the BEAST2 model configuration and a Microsoft Excel file to specify alignment files.

BEAST2 is also associated with Tracer (Rambaut & Drummond 2007) and DensiTree (Bouckaert & Heled 2014). 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 BEAST2 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 simultaneous display of many phylogenies.

However, for exploring the output of many BEAST2 runs, one would like a script to collect all parameters’ ESSes, parameter traces and posterior phylogenies. There is no single package that offers a complete solution, but examples of R packages that offer a partial solution are **BEASTmasterR**, **rBEAST** (Ratmann 2015) and **RBeast** (Faria & Suchard 2015). **BEASTmasterR** is a partial BEAUti alternative, does not offer a way to call BEAST2, nor allows for the same data analysis as done by Tracer. **RBeast** provides some plotting options and parsing of BEAST2 output files. **rBEAST** is a package developed around testing a biological hypothesis, that does not aim to be a generally used package.

Here, we present **babette**: BEAUti 2, BEAST2 and Tracer for R, which creates BEAST2 (v.2.4.7) 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 based 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.

76 2 Description

77 **babette** is written in the R programming language (R Core Team 2013) and en-
78 ables the full BEAST2 workflow from an R function call, in a similar way to what
79 BEAUti, DensiTree and Tracer do. **babette**'s main function is `run_beast2`,
80 which configures BEAST2, runs it and parses its output. `run_beast2` needs
81 at least the name of a FASTA file containing a DNA alignment. The default
82 settings for the other arguments of `run_beast2` are identical to BEAUti's and
83 BEAST2's default settings. Per alignment, a site model, clock model and tree
84 prior can be chosen. Multiple alignments can be used, each with its own (un-
85 linked) site model, clock model and tree prior.

86 **babette** currently has 61 exported functions to set up a BEAST2 config-
87 uration file. **babette** can currently handle a majority of BEAUti use cases.
88 Because of BEAUti's high number of plugins, **babette** uses a software archi-
89 tecture that is designed to be extended. Furthermore, **babette** has 7 exported
90 functions to run and help run BEAST2. One function is used to run BEAST2,
91 others allow the user to check if a BEAST2 configuration file is indeed valid.
92 Finally, **babette** has 20 exported function to parse the BEAST2 output files
93 and analyze the created posterior. **babette** gives the same ESSes and summary
94 statistics as Tracer. The data is formatted such that it can easily be visualized
95 using `ggplot2` (for a trace, similar to Tracer) or `phangorn` (Schliep 2011) (for
96 the phylogenies in a posterior, similar to DensiTree).

97 Currently, **babette** does not replace all functionality in BEAUti, as it does
98 not provide 3 out of 7 tree priors, nor does it support RNA alignments or
99 use of morphological data. The many plug-ins of BEAUti are not yet sup-
100 ported by **babette**. **babette** does not support all command-line arguments of
101 BEAST2, does not provide the more specialized Tracer analysis options, nor is it
102 as feature-rich in plotting options as DensiTree. Up until now, the **babette** fea-

103 tures implemented are those requested by users. Further extension of **babette**
104 will be based on future user requests.

105 **3 Usage**

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

```
107 library(babette)
```

108 BEAUti, and likewise **babette**, needs at least a FASTA filename to produce a
109 BEAST2 configuration file. In BEAUti, this is achieved by loading a FASTA file,
110 then saving an output file using a common save file dialog. After this, BEAST2
111 needs to be applied to the created configuration file. It creates multiple files
112 storing the posterior. These output files must be parsed by either Tracer of
113 DensiTree. In **babette**, all this is achieved by:

```
114 out <- run_beast2("anthus_aco.fas")
```

115 This code will create a (temporary) BEAST2 configuration file, from the FASTA
116 file with name **anthus_aco.fas** (which is supplied with the package, from
117 (Van Els & Norambuena 2018)), using the same default settings as BEAUti,
118 which are, among others, a Jukes-Cantor site model, a strict clock, and a Yule
119 birth tree prior. **babette** will then execute BEAST2 using that file, and parses
120 the output. The returned data structure, named **out**, is a list of parameter
121 estimates (called **estimates**), posterior phylogenies (called **anthus_aco_trees**,
122 named after the alignment's name) and MCMC operator performance (**operators**).
123 An example of using a different site model, clock model and tree prior is:

```
124 out <- run_beast2(  
125   "anthus_aco.fas",  
126   site_models = create_hky_site_model(),  
127   clock_models = create_rln_clock_model(),
```

```

128     tree_priors = create_bd_tree_prior()
129 )

```

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:

```

138 out <- run_beast2(
139   c("anthus_aco.fas", "anthus_nd2.fas"),
140   site_models = list(
141     create_tn93_site_model(),
142     create_gtr_site_model()
143   )
144 )

```

babette also uses the same default prior distributions as BEAUti for each of the site models, clock models and tree priors. For example, by default, 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 (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. To do this in **babette**:

```

153 out <- run_beast2(
154   "anthus_aco.fas",

```

```

155     tree_priors = create_yule_tree_prior(
156         birth_rate_distr = create_exp_distr()
157     )
158 )

```

159 In this same example, one may specify the initial shape parameters of the expo-
160 nential distribution. In BEAST2's implementation, an exponential distribution
161 has one shape parameter: its mean, which can be set to any value with BEAUti.
162 Within **babette**, to set the mean value of the exponential distribution to a fixed
163 (non-estimated) value, do:

```

164 out <- run_beast2(
165     "anthus_aco.fas",
166     tree_priors = create_yule_tree_prior(
167         birth_rate_distr = create_exp_distr(
168             mean = create_mean_param(
169                 value = 1.0,
170                 estimate = FALSE
171             )
172         )
173     )
174 )

```

175 Our initial motivation to create **babette** was that we wanted to fix the crown
176 age of a phylogeny. BEAUti assumes that a phylogeny has a crown age that
177 needs to be jointly estimated with the phylogeny and other parameters. It does
178 not allow for fixing the crown age. Without **babette**, one needs to manually edit
179 the BEAST2 configuration file (Bouckaert & Vaughan 2017), which is tedious
180 and prone to errors. Fixing the crown ages is especially useful for theoretical
181 experiments, as this allows for one less source of variation. This is how to specify
182 a fixed crown age with **babette**:

```

183 out <- run_beast2(
184   "anthus_aco.fas",
185   posterior_crown_age = 15
186 )

```

187 **babette** allows for the same functionality as Tracer. Tracer works on the values
188 of the parameter estimates sampled in the BEAST2 run. This is called the
189 "trace" (hence the name). The start of the trace is usually discarded, as an
190 MCMC algorithm (such as used by BEAST2) first has to converge to its equi-
191 librium. The start of the trace, called the "burn-in", will be removed, because
192 its parameter estimates are not representative. By default, Tracer discards the
193 first 10% of all the parameter estimates. To remove a 20% burn-in from all
194 parameter estimates in **babette**, the following code can be used:

```

195 traces <- remove_burn_ins(
196   out$estimates,
197   burn_in_fraction = 0.2
198 )

```

199 Tracer shows the ESSes of each posterior's variables. These ESSes are important
200 to determine the strength of the inference. As a rule of thumb, an ESS of 200 is
201 acceptable for any parameter estimate. To calculate the effective sample sizes
202 (of all estimated variables) in **babette**:

```

203 esses <- calc_esses(
204   traces,
205   sample_interval = 1000
206 )

```

207 Tracer displays multiple summary statistics for each estimated variable: the
208 mean and its standard error, standard deviation, variance, median, mode, geo-
209 metric mean, 95% highest posterior density interval, auto-correlation time and

effective sample size. It displays these statistics per variable. In **babette**, these summary statistics are collected for all estimated parameters at once:

```
sum_stats <- calc_summary_stats(  
  traces,  
  sample_interval = 1000  
)
```

babette allows for the same functionality as **DensiTree**. **DensiTree** displays the phylogenies in a posterior at the same time scale, drawn one over one another, allowing to see the uncertainty in topology and branch lengths. Within the object **out**, the posterior phylogenies are stored as **anthus_aco_trees**, and can be plotted as such:

```
plot_densitree(out$anthus_aco_trees)
```

4 **babette** 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 (<https://travis-ci.org>) continuous integration service, which is known to significantly increase the number of bugs exposed (Vasilescu *et al.* 2015) and increases the speed at which new features are added (Vasilescu *et al.* 2015). **babette** has a 100% code coverage, which correlates with code quality (Horgan *et al.* 1994; Del Frate *et al.* 1995). **babette** follows Hadley Wickham's style guide (Wickham 2015), which improves software quality (Fang 2001). **babette** depends on multiple packages, which are **ape** (Paradis *et al.* 2004), **beautier** (Bilderbeek 2018b), **beastier** (Bilderbeek 2018a), **devtools** (Wickham & Chang 2016), **geiger** (Harmon *et al.* 2008), **ggplot2** (Wickham 2009),

235 `knitr` (Xie 2017), `phangorn` (Schliep 2011), `rmarkdown` (Allaire *et al.* 2017),
236 `seqinr` (Charif & Lobry 2007), `stringr` (Wickham 2017), `testit` (Xie 2014)
237 and `tracerer` (Bilderbeek 2018c). `babette` is tested to give a clean error mes-
238 sage for incorrect input, by calling `babette` one million times with random or
239 random sensible inputs, using the Peregrine high performance computer cluster.
240 The scripts to do so, are supplied with `babette`.

241 `babette`'s development takes place on GitHub, <https://github.com/richelbilderbeek/>
242 `babette`, which accommodates collaboration (Perez-Riverol *et al.* 2016) and im-
243 proves transparency (Gorgolewski & Poldrack 2016). `babette`'s GitHub facili-
244 tates feature requests and has guidelines how to do so.

245 `babette`'s documentation is extensive. All functions are documented in the
246 package's internal documentation. For quick use, each exported function shows
247 a minimal example. For easy exploration, each exported function's documen-
248 tation links to related functions. Additionally, `babette` has a vignette that
249 demonstrates extensively how to use it. There is documentation on the GitHub
250 to get started, with a dozen examples of BEAUti screenshots with equivalent
251 `babette` code. Finally, `babette` has some tutorial video's that can be down-
252 loaded or viewed on YouTube, <https://goo.gl/weKaaU>.

253 5 Citation of `babette`

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

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

258 6 Acknowledgements

259 Thanks to Yacine Ben Chehida and Paul van Els for supplying their BEAST2
260 use cases. Thanks again to Paul van Els for the sharing his FASTA files for use by
261 this package. Thanks to Leonel Herrera-Alsina, Raphael Scherrer and Giovanni
262 Laudanno for their comments on this package and article. We would like to
263 thank the Center for Information Technology of the University of Groningen
264 for their support and for providing access to the Peregrine high performance
265 computing cluster.

266 References

- 267 Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wick-
268 ham, H., Cheng, J. & Chang, W. (2017) *rmarkdown: Dynamic Documents*
269 *for R*. R package version 1.8.
- 270 Bilderbeek, R.J. (2018a) beastier: BEAST2 from R. [https://github.com/](https://github.com/richelbilderbeek/beastier)
271 [richelbilderbeek/beastier](https://github.com/richelbilderbeek/beastier) [Accessed: 2018-02-28].
- 272 Bilderbeek, R.J. (2018b) beautier: BEAUti 2 from R. [https://github.com/](https://github.com/richelbilderbeek/beautier)
273 [richelbilderbeek/beautier](https://github.com/richelbilderbeek/beautier) [Accessed: 2018-02-28].
- 274 Bilderbeek, R.J. (2018c) tracerer: Tracer from R. [https://github.com/](https://github.com/richelbilderbeek/tracerer)
275 [richelbilderbeek/tracerer](https://github.com/richelbilderbeek/tracerer) [Accessed: 2018-02-28].
- 276 Bouckaert, R. & Heled, J. (2014) Densitree 2: Seeing trees through the forest.
277 *bioRxiv*, p. 012401.
- 278 Bouckaert, R., Heled, J., Kühnert, D., Vaughan, T., Wu, C.H., Xie, D., Suchard,
279 M.A., Rambaut, A. & Drummond, A.J. (2014) Beast 2: a software platform
280 for bayesian evolutionary analysis. *PLoS Comput Biol*, **10**, e1003537.

281 Bouckaert, R. & Vaughan, T. (2017) Fix starting tree. [http://www.beast2.](http://www.beast2.org/fix-starting-tree)
282 [org/fix-starting-tree](http://www.beast2.org/fix-starting-tree) [Accessed: 2018-02-28].

283 Charif, D. & Lobry, J. (2007) SeqinR 1.0-2: a contributed package to the R
284 project for statistical computing devoted to biological sequences retrieval and
285 analysis. U. Bastolla, M. Porto, H. Roman & M. Vendruscolo, eds., *Struc-*
286 *tural approaches to sequence evolution: Molecules, networks, populations*, Bi-
287 ological and Medical Physics, Biomedical Engineering, pp. 207–232. Springer
288 Verlag, New York. ISBN : 978-3-540-35305-8.

289 Del Frate, F., Garg, P., Mathur, A.P. & Pasquini, A. (1995) On the correlation
290 between code coverage and software reliability. *Software Reliability Engi-*
291 *neering, 1995. Proceedings., Sixth International Symposium on*, pp. 124–132.
292 IEEE.

293 Drummond, A.J., Suchard, M.A., Xie, D. & Rambaut, A. (2012) Bayesian phy-
294 logenetics with beauti and the beast 1.7. *Molecular biology and evolution*, **29**,
295 1969–1973.

296 Fang, X. (2001) Using a coding standard to improve program quality. *Quality*
297 *Software, 2001. Proceedings. Second Asia-Pacific Conference on*, pp. 73–78.
298 IEEE.

299 Faria, N. & Suchard, M.A. (2015) RBeast. [https://github.com/beast-dev/](https://github.com/beast-dev/RBeast)
300 [RBeast](https://github.com/beast-dev/RBeast) [Accessed: 2018-03-02].

301 Gorgolewski, K.J. & Poldrack, R. (2016) A practical guide for improving trans-
302 parency and reproducibility in neuroimaging research. *bioRxiv*, p. 039354.

303 Harmon, L., Weir, J., Brock, C., Glor, R. & Challenger, W. (2008) Geiger:
304 investigating evolutionary radiations. *Bioinformatics*, **24**, 129–131.

305 Horgan, J.R., London, S. & Lyu, M.R. (1994) Achieving software quality with
306 testing coverage measures. *Computer*, **27**, 60–69.

307 Matzke, N.J. (2015) BEASTmaster: R tools for automated conversion of
308 NEXUS data to BEAST2 XML format, for fossil tip-dating and other uses.
309 <https://github.com/nmatzke/BEASTmaster> [Accessed: 2018-02-28].

310 Paradis, E., Claude, J. & Strimmer, K. (2004) APE: analyses of phylogenetics
311 and evolution in R language. *Bioinformatics*, **20**, 289–290.

312 Perez-Riverol, Y., Gatto, L., Wang, R., Sachsenberg, T., Uszkoreit, J., Lepre-
313 vost, F., Fufezan, C., Ternent, T., Eglen, S.J., Katz, D.S. *et al.* (2016) Ten
314 simple rules for taking advantage of git and github. *bioRxiv*, p. 048744.

315 R Core Team (2013) *R: A Language and Environment for Statistical Computing*.
316 R Foundation for Statistical Computing, Vienna, Austria.

317 Rambaut, A. & Drummond, A.J. (2007) *Tracer v1.4*. Available from
318 <http://beast.bio.ed.ac.uk/Tracer>.

319 Ratmann, O. (2015) rBEAST. <https://github.com/olli0601/rBEAST> [Ac-
320 cessed: 2018-03-02].

321 Schliep, K. (2011) phangorn: phylogenetic analysis in R. *Bioinformatics*, **27**,
322 592–593.

323 Van Els, P. & Norambuena, H.V. (2018) A revision of species limits in neotrop-
324 ical pipits anthus based on multilocus genetic and vocal data. *Ibis*.

325 Vasilescu, B., Yu, Y., Wang, H., Devanbu, P. & Filkov, V. (2015) Quality and
326 productivity outcomes relating to continuous integration in github. *Proceed-*
327 *ings of the 2015 10th Joint Meeting on Foundations of Software Engineering*,
328 pp. 805–816. ACM.

- 329 Wickham, H. (2009) *ggplot2: elegant graphics for data analysis*. Springer New
330 York.
- 331 Wickham, H. (2015) *R packages: organize, test, document, and share your code*.
332 O'Reilly Media, Inc.
- 333 Wickham, H. (2017) *stringr: Simple, Consistent Wrappers for Common String*
334 *Operations*. R package version 1.2.0.
- 335 Wickham, H. & Chang, W. (2016) *devtools: Tools to Make Developing R Pack-*
336 *ages Easier*. R package version 1.12.0.9000.
- 337 Xie, Y. (2014) *testit: A Simple Package for Testing R Packages*. R package
338 version 0.4, <http://CRAN.R-project.org/package=testit>.
- 339 Xie, Y. (2017) *knitr: A General-Purpose Package for Dynamic Report Genera-*
340 *tion in R*. R package version 1.17.

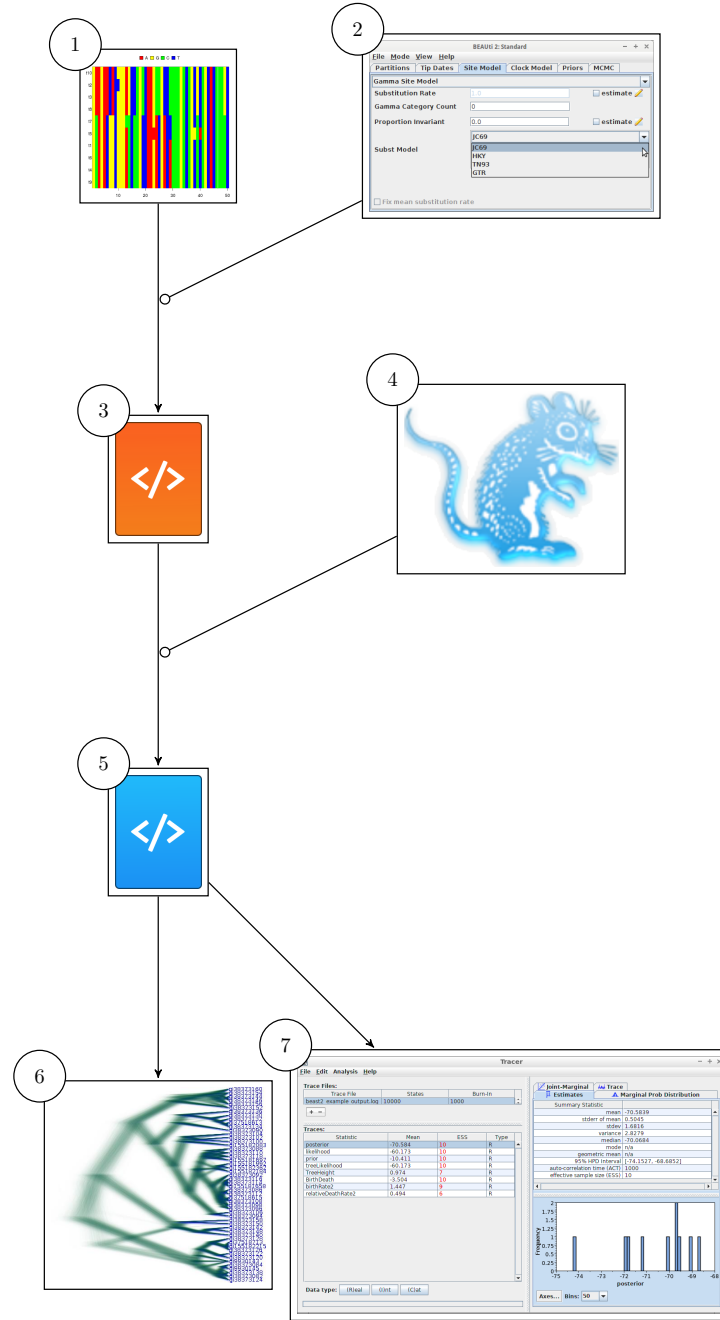


Figure 1: Workflow using GUI tools. From an alignment (1) and BEAUti (2), a BEAST2 configuration file (3) is created. BEAST2 (4) uses that file to infer a posterior, storing it in multiple files (5). These results are visualized using DensiTree (6) and Tracer (7). babette allows for the same workflow,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

Name	Description
run_beast2	Run BEAST2
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