

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 July 1, 2019

Summary

1. In the field of phylogenetics, BEAST2 is one of the most widely used software tools. It comes with the graphical user interfaces BEAUti 2, DensiTree and Tracer, to create BEAST2 configuration files and to interpret BEAST2's output files. However, when many different alignments or model setups are required, a workflow of graphical user interfaces 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 and the novel functionality it provides compared to the original tools and we 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.

21

22

Samenvatting

1. In de fylogenetica is BEAST2 een van de meest gebruikte hulpprogramma's. Het is gebundeld met de grafische gebruikersinterface BEAUti 2, DensiTree en Tracer, om BEAST2-configuratiebestanden te maken en om BEAST2-outputbestanden te interpreteren. Echter, als veel verschillende aligneringen of modelopzetten nodig zijn, is een werkvolgorde van meerdere grafische gebruikersinterfaces onhandig.

2. Hier presenteren we een gratis, vrij en open-source package, **babette**: 'BEAUti 2, BEAST2 en Tracer voor R', voor de programmeertaal R. **babette** schrijft BEAST2-configuratiebestanden, start BEAST2 and verwerkt de resultaten, alles met een enkele R functie-aanroep.

3. We beschrijven hoe **babette** te gebruiken is en de nieuwe mogelijkheden die het biedt vergeleken met de originele programma's, aan de hand

van enkele voorbeelden.

4. Omdat **babette** ontworpen is voor uitbreidbaarheid en hoge kwaliteit, sluiten we af met het beschrijven van de verdere ontwikkeling van dit 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) are related to each other, but we can also estimate relevant parameters such as extinction and speciation rates from them. There are many phylogenetics tools available to obtain an estimate of the phylogeny of a given set of species. BEAST2 (Bouckaert *et al.* 2014) is one of the most widely used ones. It uses a Bayesian statistical framework to estimate the joint posterior distribution of phylogenies and model parameters, from one or more DNA, RNA or amino acid alignments (see figure 1 for an overview of the workflow).

BEAST2 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 align-

61 ments and doing a superficial analysis of sensitivity of the reconstructed tree to
62 model settings. For exploring many trees (for instance from simulations), for a
63 sliding-window analysis on a genomic alignment, or for a more thorough sensi-
64 tivity analysis, one would like to loop through multiple (simulated or shortened)
65 alignments, nucleotide substitution models, clock models and tree priors. One
66 such tool to replace BEAUti is **BEASTmasterR** (Matzke 2015), which focuses on
67 morphological traits and tip-dating, but also supports DNA data. **BEASTmasterR**,
68 however, requires hundreds of lines of R code to setup the BEAST2 model con-
69 figuration and a Microsoft Excel file to specify alignment files.

70 BEAST2 is also associated with Tracer (Rambaut & Drummond 2007) and
71 DensiTree (Bouckaert & Heled 2014). Both are desktop applications to an-
72alyze the output of BEAST2, each with a user-friendly graphical user inter-
73 face. Tracer’s purpose is to analyze the parameter estimates generated from
74 a (BEAST1 and) BEAST2 run. It shows, among others, the effective sample
75 size (ESS) and time series (‘the trace’, hence the name) of each variable in the
76 MCMC run. Both ESS and trace are needed to assess the strength of the infer-
77 ence. DensiTree visualizes the phylogenies of a BEAST2 posterior, with many
78 options to improve the simultaneous display of many phylogenies.

79 However, for exploring the output of many BEAST2 runs, one would like a
80 script to collect all parameters’ ESSes, parameter traces and posterior phyloge-
81 nies. There is no single package that offers a complete solution, but examples
82 of R packages that offer a partial solution are rBEAST (Ratmann 2015) and
83 RBeast (Faria & Suchard 2015). RBeast provides some plotting options and
84 parsing of BEAST2 output files, but the plotting functions are too specific for
85 general use. rBEAST was developed to test a particular biological hypothesis
86 (Ratmann *et al.* 2016), and hence was not designed for general use.

87 Here, we present **babette**: ‘BEAUti 2, BEAST2 and Tracer for R’, which

88 creates BEAST2 (v.2.4.7) configuration files, runs BEAST2, and analyzes its
89 results, all from an R function call. This will save time, tedious mouse clicking
90 and reduces the chances of errors in such repetitive actions. The interface of
91 **babette** mimics the tools it is based on. This familiarity helps both beginner
92 and experienced BEAST2 users to make the step from those tools to **babette**.
93 **babette** enables the creation of a single-script pipeline from sequence alignments
94 to posterior analysis in R.

95 2 Description

96 **babette** is written in the R programming language (R Core Team 2013) and
97 enables the full BEAST2 workflow from a single R function call, in a similar
98 way to what subsequent usage of BEAUti, DensiTree and Tracer would produce.
99 **babette**'s main function is **bbt_run**, which configures BEAST2, runs it and
100 parses its output. **bbt_run** needs at least the name of a FASTA file containing
101 a DNA alignment. The default settings for the other arguments of **bbt_run**
102 are identical to BEAUti's and BEAST2's default settings. Per alignment, a site
103 model, clock model and tree prior can be chosen. Multiple alignments can be
104 used, each with its own (unlinked) site model, clock model and tree prior.

105 **babette** currently has 108 exported functions to set up a BEAST2 config-
106 uration file. **babette** can currently handle the majority of BEAUti use cases.
107 Because of BEAUti's high number of plugins, **babette** uses a software architec-
108 ture that is designed to be extended. Furthermore, **babette** has 13 exported
109 functions to run and help run BEAST2. One function is used to run BEAST2,
110 another one installs BEAST2 to a default location. Finally, **babette** has 21
111 exported function to parse the BEAST2 output files and analyze the created
112 posterior. **babette** gives the same ESSes and summary statistics as Tracer.
113 The data is formatted such that it can easily be visualized using **ggplot2** (for

114 a trace, similar to Tracer) or **phangorn** (Schliep 2011) (for the phylogenies in a
115 posterior, similar to DensiTree).

116 Currently, **babette** does not contain all functionality in BEAUti, BEAST2
117 and their many plug-ins, because these tools themselves also change in time.
118 **babette** currently works only on DNA data, because this is the most common
119 use case. Nevertheless, **babette** provides the majority of default tree priors and
120 supports the most important command-line arguments of BEAST2, provides the
121 core Tracer analysis options, and has the most basic subset of plotting options of
122 DensiTree. Up till now, the **babette** features implemented are those requested
123 by users. Further extension of **babette** will be based on future user requests.

124 3 Usage

125 **babette** can be installed easily from CRAN:

126

```
127 install.packages("babette")
```

128 For the most up-to-date version, one can download and install the package from
129 **babette**'s GitHub repository:

130

```
131 devtools::install_github("richelbilderbeek/babette")
```

132 To start using **babette**, load its functions in the global namespace first:

133

```
134 library(babette)
```

135 Because **babette** calls BEAST2, BEAST2 must be installed. This can be done
136 from R, using:

137

```
138 install_beast2()
```

139 This will install BEAST2 to the default user data folder, but a different path
140 can be specified as well. BEAUti, and likewise **babette**, needs at least a FASTA
141 filename to produce a BEAST2 configuration file. In BEAUti, this is achieved
142 by loading a FASTA file, then saving an output file using a common save file
143 dialog. After this, BEAST2 needs to be applied to the created configuration
144 file. It creates multiple files storing the posterior. These output files must be
145 parsed by either Tracer or DensiTree. In **babette**, all this is achieved by:

```
146
```

```
147 out <- bbt_run(fasta_filenames = "anthus_aco.fas")
```

148 This code will create a (temporary) BEAST2 configuration file, from the FASTA
149 file with name **anthus_aco.fas** (which is supplied with the package, from
150 Van Els & Norambuena 2018), using the same default settings as BEAUti, which
151 are, among others, a Jukes-Cantor site model, a strict clock, and a Yule birth
152 tree prior. **babette** will then execute BEAST2 using that file, and parses the
153 output. The returned data structure, named **out**, is a list of parameter estimates
154 (called **estimates**), posterior phylogenies (called **anthus_aco_trees**, named af-
155 ter the alignment's name) and MCMC operator performance (**operators**). An
156 example of using a different site model, clock model and tree prior is:

```
157
```

```
158 out <- bbt_run(  
159   fasta_filenames = "anthus_aco.fas",  
160   site_models = create_hky_site_model(),  
161   clock_models = create_rln_clock_model(),  
162   tree_priors = create_bd_tree_prior()  
163 )
```

164 This code uses an HKY site model, a relaxed log-normal clock model and a birth-
 165 death tree prior, each with their default settings in BEAUti. Table 1 shows an
 166 overview of all functions to create site models, clock models and tree priors. Note
 167 that the arguments' names `site_models`, `clock_models` and `tree_priors` are
 168 plural, as each of these can be (a list of) one or more elements. Each of these
 169 arguments must have the same number of elements, so that each alignment has
 170 its own site model, clock model and tree prior. An example of two alignments,
 171 each with its own site model, is:

172

```
173 out <- bbt_run(  
174   fasta_filenames = c(  
175     "anthus_aco.fas",  
176     "anthus_nd2.fas"  
177   ),  
178   site_models = list(  
179     create_tn93_site_model(),  
180     create_gtr_site_model()  
181   )  
182 )
```

183 **babette** also uses the same default prior distributions as BEAUti for each of
 184 the site models, clock models and tree priors. For example, by default, a Yule
 185 tree prior assumes that the birth rate follows a uniform distribution, from minus
 186 infinity to plus infinity. One may prefer a different distribution instead. Here
 187 is an example how to specify an exponential distribution for the birth rate in a
 188 Yule tree prior in **babette**:

189

```
190 out <- bbt_run(  
191   fasta_filenames = "anthus_aco.fas",
```



```

192     tree_priors = create_yule_tree_prior(
193         birth_rate_distr = create_exp_distr()
194     )
195 )

```

196 In this same example, one may specify the initial shape parameters of the expo-
197 nential distribution. In BEAST2's implementation, an exponential distribution
198 has one shape parameter: its mean, which can be set to any value with BEAUti.
199 To set the mean value of the exponential distribution to a fixed (non-estimated)
200 value, do:

```

201
202 out <- bbt_run(
203     fasta_filenames = "anthus_aco.fas",
204     tree_priors = create_yule_tree_prior(
205         birth_rate_distr = create_exp_distr(
206             mean = create_mean_param(
207                 value = 1.0,
208                 estimate = FALSE
209             )
210         )
211     )
212 )

```

213 **babette** also supports node dating. Like BEAUti, one can specify Most Recent
214 Common Ancestor ('MRCA') priors. An MRCA prior allows to specify taxa
215 having a common ancestor, including a distribution for the date of that ancestor.
216 With **babette**, this is achieved as follows:

```

217
218 out <- bbt_run(
219     fasta_filenames = "anthus_aco.fas",

```

```

220   mrca_priors = create_mrca_prior(
221     taxa_names = sample(get_taxa_names("anthus_aco.fas"),
222       size = 2),
223     alignment_id = get_alignment_id("anthus_aco.fas"),
224     is_monophyletic = TRUE,
225     mrca_distr = create_normal_distr(
226       mean = create_mean_param(value = 15.0, estimate =
227         FALSE),
228       sigma = create_sigma_param(value = 0.025, estimate =
229         FALSE)
230     )
231   )
232 )

```

233 Instead of dating the ancestor of two random taxa, any subset of taxa can
 234 be selected, and multiple sets are allowed. **babette** allows for the same core
 235 functionality as Tracer to show the values of the parameter estimates sampled
 236 in the BEAST2 run. This is called the "trace" (hence the name). The start
 237 of the trace, called the "burn-in", is usually discarded, as an MCMC algorithm
 238 (such as used by BEAST2) first has to converge to its equilibrium and hence
 239 the parameter estimates are not representative. By default, Tracer discards the
 240 first 10% of all the parameter estimates. To remove a 20% burn-in from all
 241 parameter estimates in **babette**, the following code can be used:

```

242
243   traces <- remove_burn_ins(
244     traces = out$estimates,
245     burn_in_fraction = 0.2
246   )

```

247 Tracer shows the ESSes of each posterior's variables. These ESSes are important

248 to determine the strength of the inference. As a rule of thumb, an ESS of 200 is
249 acceptable for any parameter estimate. To calculate the effective sample sizes
250 (of all estimated variables) in **babette**:

251

```
252 esses <- calc_esses(  
253   traces = traces,  
254   sample_interval = 1000  
255 )
```

256 Tracer displays multiple summary statistics for each estimated variable: the
257 mean and its standard error, standard deviation, variance, median, mode, geo-
258 metric mean, 95% highest posterior density interval, auto-correlation time and
259 effective sample size. It displays these statistics per variable. In **babette**, these
260 summary statistics are collected for all estimated parameters at once:

261

```
262 sum_stats <- calc_summary_stats(  
263   traces = traces,  
264   sample_interval = 1000  
265 )
```

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

271

```
272 plot_densitree(phylos = out$anthus_aco_trees)
```

273 Instead of running the full pipeline, **babette** also allows to only create a BEAST2
274 configuration file. To create a BEAST2 configuration file, with all settings to

275 default, use:

276

```
277 create_beast2_input_file(  
278   input_filenames = babette::get_babette_path("anthus_aco.  
279     fas"),  
280   output_filename = "beast2.xml"  
281 )
```

282 This file can then be loaded and edited by BEAUti, run by BEAST2, or run by

283 **babette**:

284

```
285 run_beast2(  
286   input_filename = "beast2.xml",  
287   output_log_filename = "run.log",  
288   output_trees_filenames = "posterior.trees",  
289   output_state_filename = "final.xml.state"  
290 )
```

291 **run_beast2** is a function that only runs BEAST2, and does not parse the output
292 files (unlike **bbt_run**). In the example above, we specify the names of the desired
293 BEAST2 output files explicitly, and these will be created in the R working
294 directory, after which they can be inspected with other tools, or used to continue
295 a BEAST2 run. When the names of these files are not specified, both **bbt_run**
296 and **run_beast2** put these files in the default temporary folder (as obtained
297 from **temp.dir()**) to keep the working directory clean of intermediate files.

298 4 **babette** resources

299 **babette** is free, libre and open source software available at [http://github.](http://github.com/richelbilderbeek/babette)
300 [com/richelbilderbeek/babette](http://github.com/richelbilderbeek/babette) and is licensed under the GNU General Pub-

301 lic License v3.0. **babette** uses the Travis CI (<https://travis-ci.org>) con-
302 tinuous integration service, which is known to significantly increase the number
303 of bugs exposed (Vasilescu *et al.* 2015) and increases the speed at which new
304 features are added (Vasilescu *et al.* 2015). **babette** has a 100% code cover-
305 age, which correlates with code quality (Horgan *et al.* 1994; Del Frate *et al.*
306 1995). **babette** follows Hadley Wickham’s style guide (Wickham 2015), which
307 improves software quality (Fang 2001). **babette** depends on multiple packages,
308 which are **ape** (Paradis *et al.* 2004), **beautier** (Bilderbeek 2018b), **beastier**
309 (Bilderbeek 2018a), **devtools** (Wickham & Chang 2016), **geiger** (Harmon *et al.*
310 2008), **ggplot2** (Wickham 2009), **knitr** (Xie 2017), **phangorn** (Schliep 2011),
311 **rmarkdown** (Allaire *et al.* 2017), **seqinr** (Charif & Lobry 2007), **stringr** (Wick-
312 ham 2017), **testit** (Xie 2014) and **tracerer** (Bilderbeek 2018c). We tested
313 **babette** to give a clean error message for incorrect input, by calling **babette**
314 one million times with random or random sensible inputs, using a high perfor-
315 mance computer cluster. The test scripts are supplied with **babette**.

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

320 **babette**’s documentation is extensive. All functions are documented in the
321 package’s internal documentation. For quick use, each exported function shows
322 a minimal example. For easy exploration, each exported function’s documen-
323 tation links to related functions. Additionally, **babette** has a vignette that
324 demonstrates extensively how to use it. There is documentation on the GitHub
325 to get started, with a dozen examples of BEAUti screenshots with equivalent
326 **babette** code. Finally, **babette** has tutorial videos that can be downloaded or
327 viewed on YouTube, <https://goo.gl/weKaaU>.

328 5 Citation of babette

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

332

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

334 6 Acknowledgements

335 Thanks to Yacine Ben Chehida and Paul van Els for supplying their BEAST2
336 use cases. Thanks again to Paul van Els for sharing his FASTA files for use by
337 this package. Thanks to Leonel Herrera-Alsina, Raphael Scherrer and Giovanni
338 Laudanno for their comments on this package and article. Thanks to Huw
339 Ogilvie, Michael Matschiner and one anonymous reviewer for reviewing this
340 article. Thanks to rOpenSci, and especially Noam Ross and Guangchuang Yu
341 for reviewing the package's source code. We would like to thank the Center for
342 Information Technology of the University of Groningen for their support and
343 for providing access to the Peregrine high performance computing cluster. We
344 thank the Netherlands Organization for Scientific Research (NWO) for financial
345 support through a VICI grant awarded to RSE.

346 7 Data Accessibility

347 All code is archived at http://github.com/richelbilderbeek/babette_article,
348 with DOI <https://doi.org/10.5281/zenodo.1251203>.

349 8 Authors' contributions

350 RJCB and RSE conceived the idea for the package. RJCB created and tested
351 the package, and wrote the first draft of the manuscript. RSE contributed
352 substantially to revisions.

353 References

- 354 Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wick-
355 ham, H., Cheng, J. & Chang, W. (2017) *rmarkdown: Dynamic Documents*
356 *for R*. R package version 1.8.
- 357 Bilderbeek, R.J. (2018a) beastier: BEAST2 from R. [https://github.com/](https://github.com/richelbilderbeek/beastier)
358 [richelbilderbeek/beastier](https://github.com/richelbilderbeek/beastier) [Accessed: 2018-03-16].
- 359 Bilderbeek, R.J. (2018b) beautier: BEAUti 2 from R. [https://github.com/](https://github.com/richelbilderbeek/beautier)
360 [richelbilderbeek/beautier](https://github.com/richelbilderbeek/beautier) [Accessed: 2018-03-16].
- 361 Bilderbeek, R.J. (2018c) tracerer: Tracer from R. [https://github.com/](https://github.com/richelbilderbeek/tracerer)
362 [richelbilderbeek/tracerer](https://github.com/richelbilderbeek/tracerer) [Accessed: 2018-03-16].
- 363 Bouckaert, R. & Heled, J. (2014) Densitree 2: Seeing trees through the forest.
364 *bioRxiv*, p. 012401.
- 365 Bouckaert, R., Heled, J., Kühnert, D., Vaughan, T., Wu, C.H., Xie, D., Suchard,
366 M.A., Rambaut, A. & Drummond, A.J. (2014) Beast 2: a software platform
367 for bayesian evolutionary analysis. *PLoS Comput Biol*, **10**, e1003537.
- 368 Charif, D. & Lobry, J. (2007) SeqinR 1.0-2: a contributed package to the R
369 project for statistical computing devoted to biological sequences retrieval and
370 analysis. U. Bastolla, M. Porto, H. Roman & M. Vendruscolo, eds., *Struc-*
371 *tural approaches to sequence evolution: Molecules, networks, populations*, Bi-

372 ological and Medical Physics, Biomedical Engineering, pp. 207–232. Springer
373 Verlag, New York. ISBN : 978-3-540-35305-8.

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

378 Drummond, A.J., Suchard, M.A., Xie, D. & Rambaut, A. (2012) Bayesian phy-
379 logenetics with beauti and the beast 1.7. *Molecular biology and evolution*, **29**,
380 1969–1973.

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

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

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

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

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

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

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

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

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

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

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

406 Ratmann, O., Van Sighem, A., Bezemer, D., Gavryushkina, A., Jurriaans, S.,
407 Wensing, A., De Wolf, F., Reiss, P., Fraser, C. *et al.* (2016) Sources of hiv
408 infection among men having sex with men and implications for prevention.
409 *Science translational medicine*, **8**, 320ra2–320ra2.

410 Schliep, K. (2011) phangorn: phylogenetic analysis in R. *Bioinformatics*, **27**,
411 592–593.

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

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

418 Wickham, H. (2009) *ggplot2: elegant graphics for data analysis*. Springer New
419 York.

420 Wickham, H. (2015) *R packages: organize, test, document, and share your code*.
421 O'Reilly Media, Inc.

422 Wickham, H. (2017) *stringr: Simple, Consistent Wrappers for Common String*
423 *Operations*. R package version 1.2.0.

424 Wickham, H. & Chang, W. (2016) *devtools: Tools to Make Developing R Pack-*
425 *ages Easier*. R package version 1.12.0.9000.

426 Xie, Y. (2014) *testit: A Simple Package for Testing R Packages*. R package
427 version 0.4, <http://CRAN.R-project.org/package=testit>.

428 Xie, Y. (2017) *knitr: A General-Purpose Package for Dynamic Report Genera-*
429 *tion in R*. R package version 1.17.

Name	Description
bbt_run	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

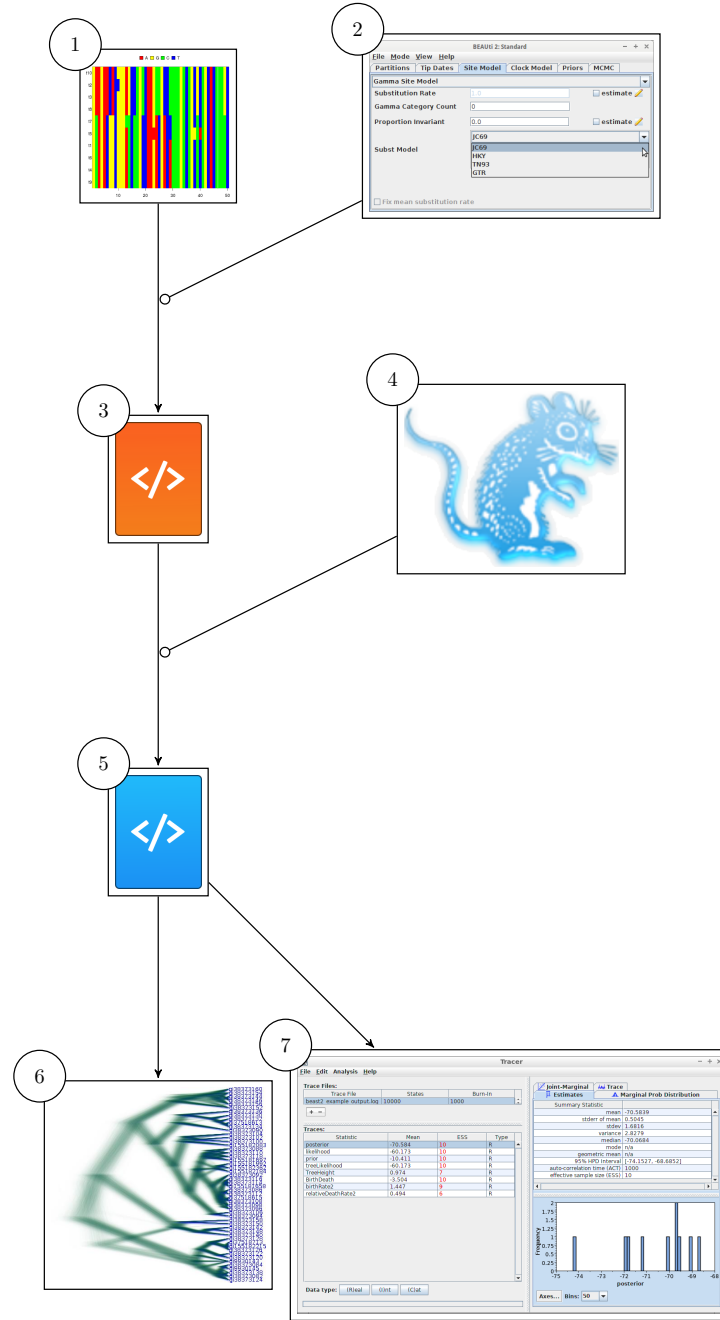


Figure 1: Workflow using GUI tools. From an alignment (1) and BEAUti (2), a BEAST2 configuration file (3) is created. BEAST2 (4) uses that file to infer a posterior, storing it in multiple files (5). These results are visualized using DensiTree (6) and Tracer (7). **babette** allows for the same workflow,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